
引脚配置工具快速入门用户指南

Document Number: PINSGS
Rev 0, 05/2016





小节编号	内容 标题	页
第 1 章 简介		
1.1	特性.....	5
1.2	约定.....	5
1.3	版本.....	6
1.4	工具本地化.....	8
第 2 章 示例用法		
第 3 章 配置		
3.1	创建新配置.....	17
3.2	保存配置.....	19
3.3	打开现有配置.....	20
第 4 章 引脚配置工具		
4.1	选择引脚配置工具.....	23
4.2	引脚路由原则.....	23
4.3	用户界面.....	24
4.3.1	封装.....	26
4.3.2	路由引脚视图.....	29
4.3.3	视图控制.....	30
4.3.4	筛选路由引脚.....	31
4.3.5	高亮显示和颜色代码.....	32
4.3.6	在引脚视图中筛选.....	34
4.3.7	函数.....	35
4.3.8	外设视图.....	37
4.3.9	引脚表.....	39
4.3.10	电源组.....	41

小节编号	标题	页
4.3.11	寄存器视图.....	42
4.4	错误和警告.....	43
4.4.1	未完成的路由.....	44
4.5	代码生成.....	45
4.5.1	导出源代码.....	47
4.5.2	导入源代码.....	48
4.6	选项.....	49
4.6.1	配置首选项.....	49
4.6.2	引脚属性.....	50
4.6.3	软件更新.....	50

第 5 章 高级功能

5.1	切换处理器.....	53
5.2	导出引脚表.....	55

第 6 章 技术支持

第 1 章 简介

引脚配置工具属于恩智浦评估套件和配置工具的一部分，这套工具能够在从初步评估至软件开发的各阶段为用户提供帮助。您可以通过本工具来创建、检测和修改器件引脚的复用以及其它配置选项。本文档为您介绍引脚配置工具。文档描述了引脚配置工具的基本组件，并列出了配置和使用工具进行引脚配置的步骤。

注

本工具提供网页版和桌面版。对于 i.MX 处理器，目前仅提供桌面版。

1.1 特性

引脚配置工具可用于：

- 配置引脚路由/多路复用
- 管理路由初始化的不同功能
- 配置引脚功能/电气属性
- 生成用于路由和功能/电气属性的代码

引脚配置工具可用于定义目标器件/板的引脚复用定义。可使用保存好的配置文件或生成的 C 文件来共享配置（通过导入/导出或通过拷贝粘帖生成的源代码）。以下章节描述引脚配置工具的部件。

注

该工具可生成将引脚路由至外设的代码，但不能用于配置外设。某些外设可能需要对引脚进行额外配置，以分配功能或通道。例如，对于有些 ADC，路由会在引脚和 ADC 外设之间提供连接。之后，您即可从 ADC 外设内部分配 ADC 通道。

1.2 约定

本文档使用以下约定。

表 1-1. 本文档使用的约定

图标	说明
	表示内容与桌面版工具有关。
	表示内容与网页版工具有关。
	表示有用的提示。

1.3 版本


这套工具统称为恩智浦配置工具。这套工具提供网页版和桌面版。

对于 Kinetis，该工具称为 Kinetis 专业工具，提供网页版和桌面版。

对于 i.MX，该工具称为 i.MX 处理器引脚配置工具，仅提供桌面版。

注

桌面版工具可连接恩智浦服务器，获取可用处理器列表。使用时，可按需检索处理器数据。

	如需在离线模式中使用桌面工具，在线时需为指定处理器创建配置。工具会将该处理器的数据存储在本地的用户文件夹中，并使用户能快速访问和离线使用。
---	---

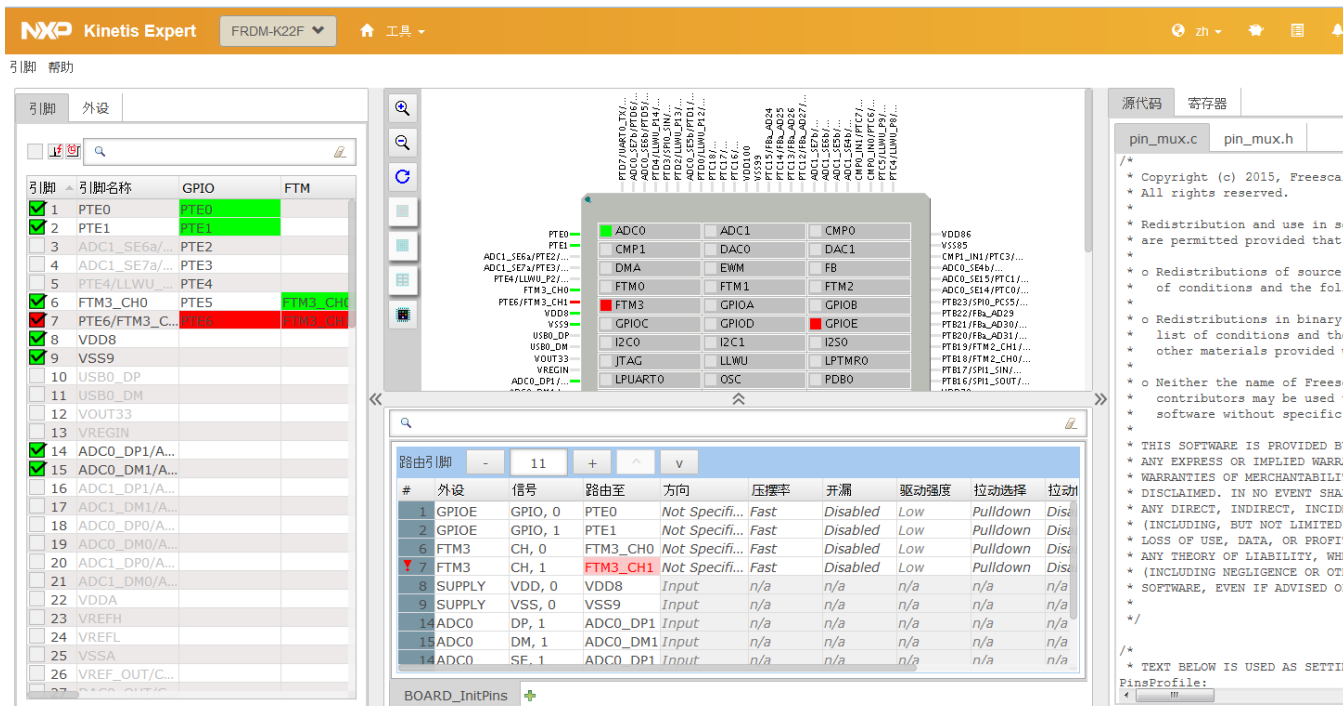


图 1-1. 网页版

 i.MX 处理器引脚配置工具仅提供桌面版。

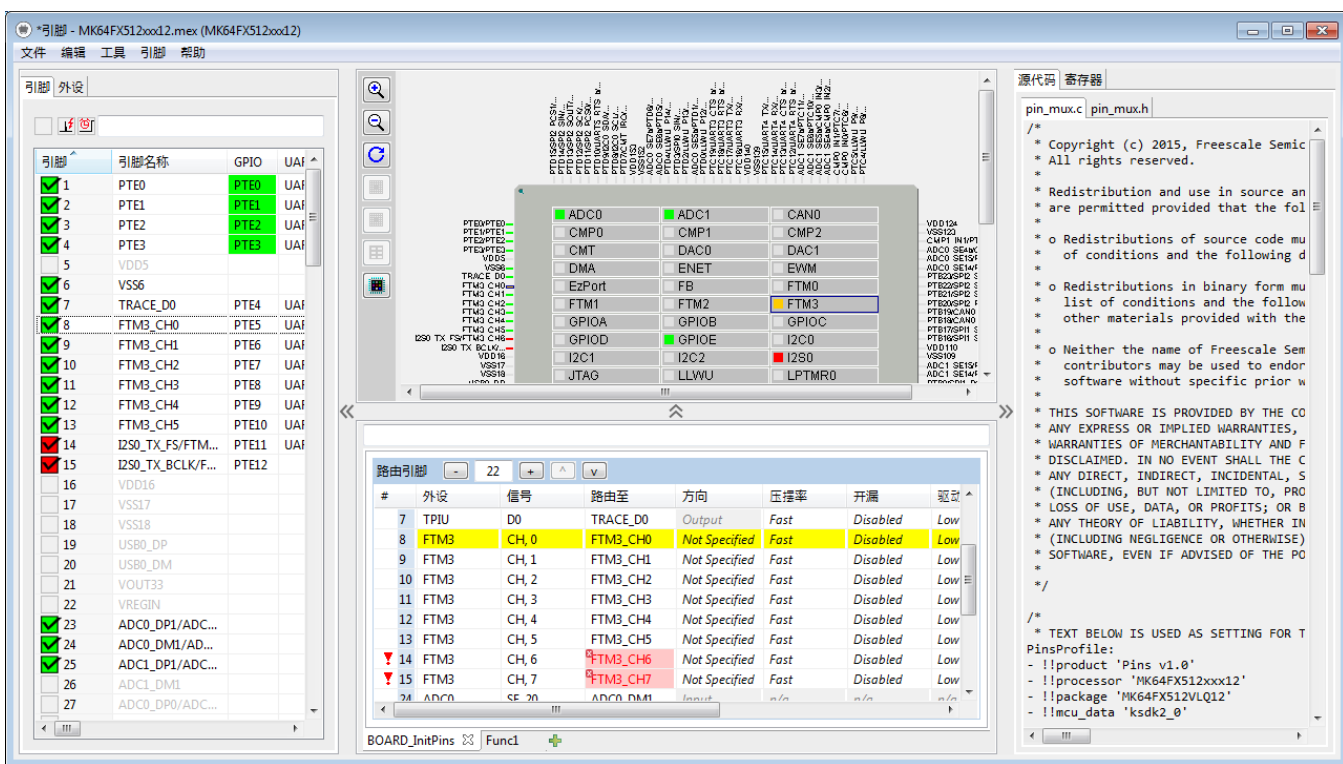


图 1-2. 桌面版

1.4 工具本地化

引脚配置工具支持中英两种语言，具体取决于您的区域设置。

要手动设置区域，请将以下参数添加到命令行：

```
tools.exe -nl zh
```

可通过在 tools.ini 文件中添加以下行来设置区域：

```
-Duser.language=zh
```

支持的语言如下：

- en - 英语
- zh - 中文

注

如果将系统区域设置为中国，工具启动时会自动显示中文菜单项、工具提示及帮助。

第 2 章 示例用法

本节列出了创建示例引脚配置（之后可在项目中使用）的步骤。

本例对电路板上的三个引脚（UART0_RX、UART0_TX 以及 PTB21）进行了配置。列出的步骤适用于桌面版和网页版。

注

仅 Kinetis 器件提供网页版工具。

您可以利用应用程序代码使用生成的文件。

1. 对于桌面版，请使用安装文件夹中的快捷方式启用工具。
2. 对于网页版，请访问 <http://kex.nxp.com> 并选择引脚配置工具。



图 2-1. 选择引脚配置工具

3. 在桌面版中，通过菜单“文件 > 新建”来创建新配置，然后选择要使用的处理器/电路板。

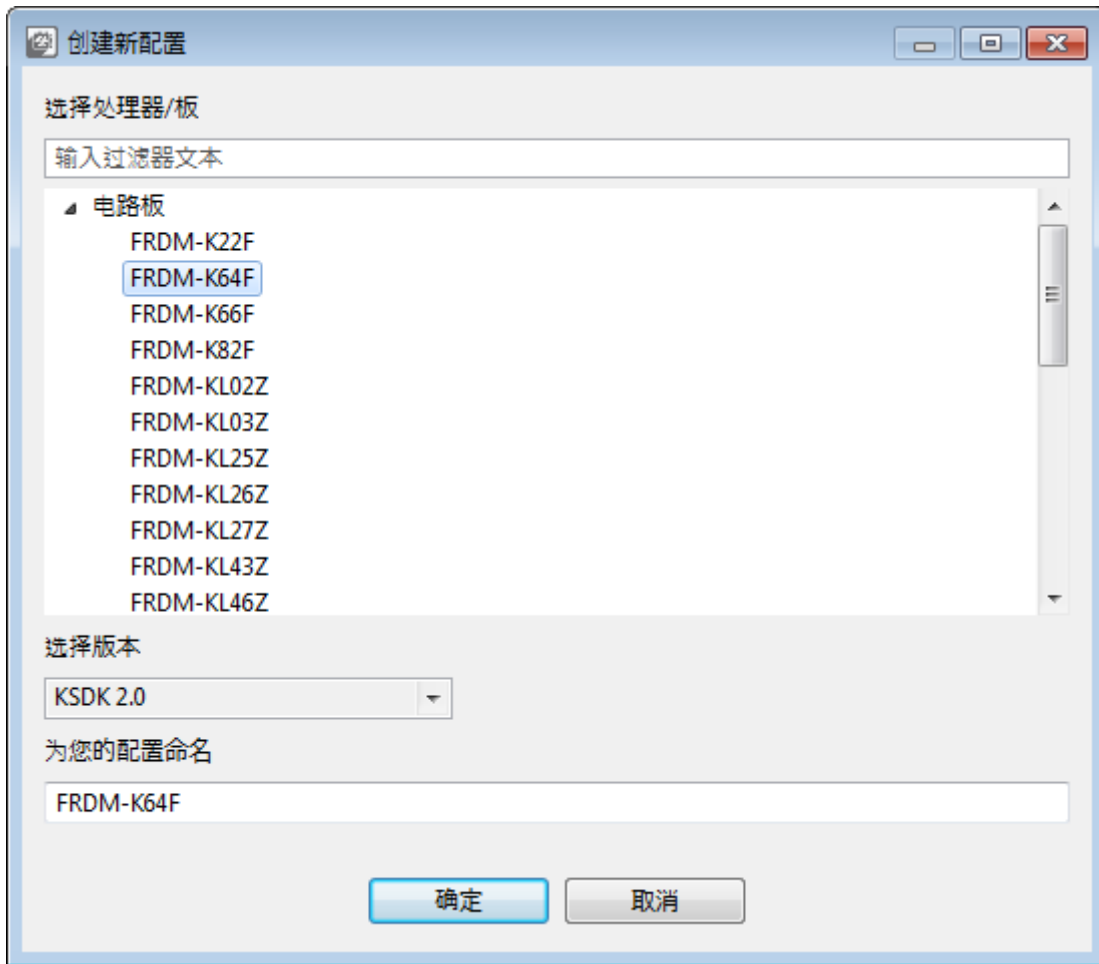


图 2-2. 创建新配置

可选择根据项目来命名配置。

4. 在网页版中:
 - a. 选择现有配置或创建新配置。

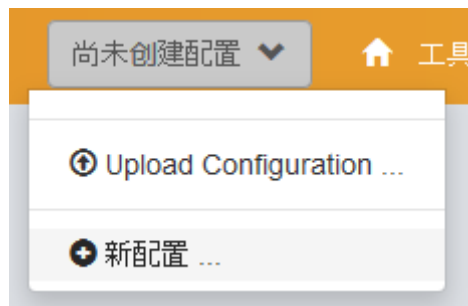


图 2-3. 新配置

- b. 要创建新配置，请选择电路板和配置。

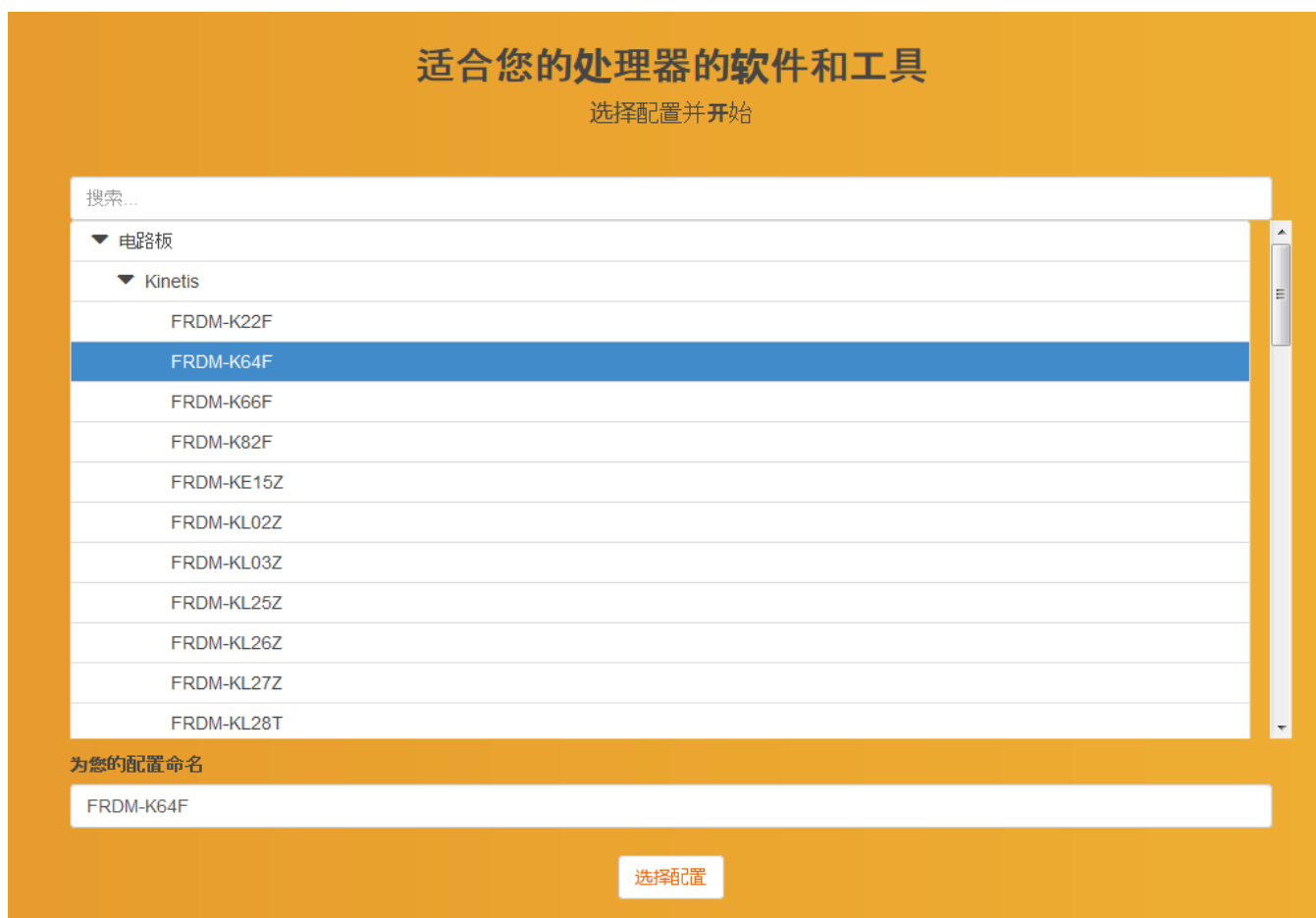


图 2-4. 选择配置并开始

c. 为配置创建 SDK 包。



图 2-5. 构建 SDK 包

d. 创建新配置后，切换至引脚配置工具。



图 2-6. 切换至引脚配置工具

5. 在左侧的”引脚“视图中，选择 UART0_RX 和 TX 信号，以及 PTB21 GPIO。为此，可单击进入单元格，将它们标成“绿色”。

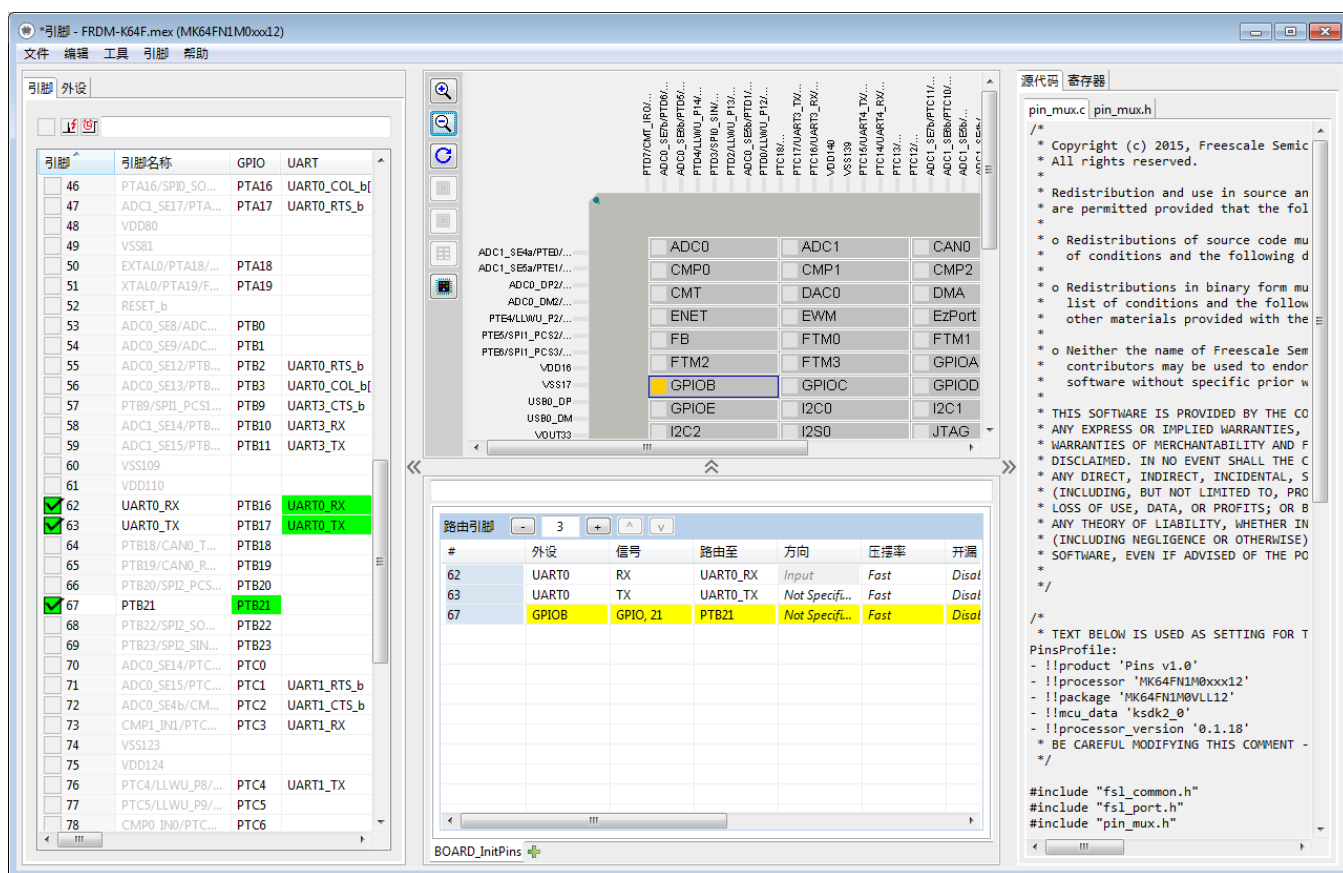


图 2-7. 切换至引脚配置工具

6. 在中间视图中，调用路由引脚表，然后为 TX 和 PTB21 信号选择 **Output**。

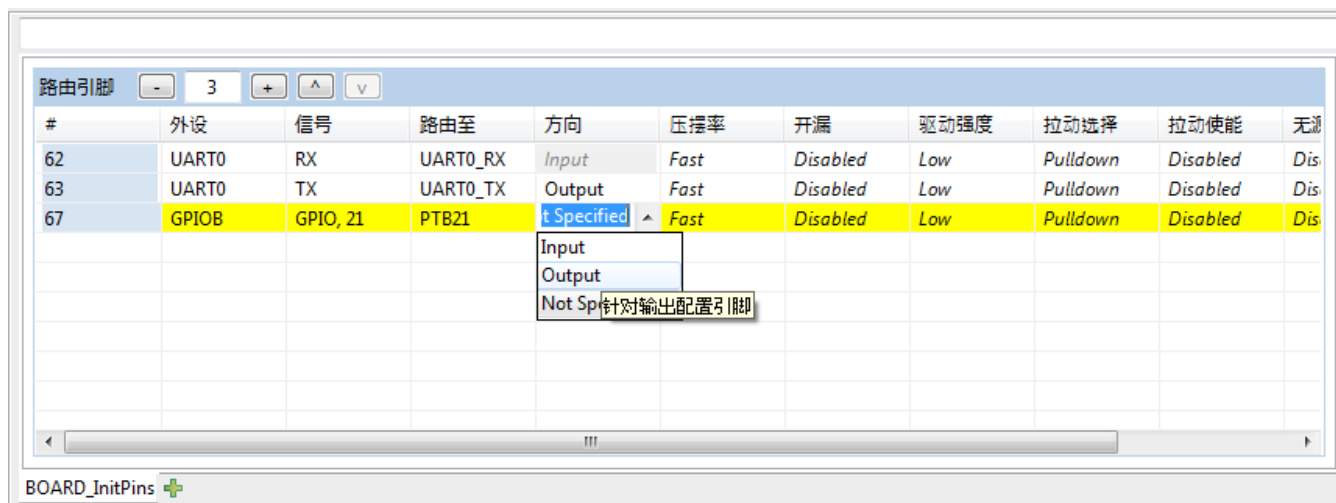


图 2-8. 选择输出

7. 引脚配置工具会自动在右侧面板上生成适用于 `pin_mux.c` 和 `pin_mux.h` 的源代码。

```

源代码 寄存器
pin_mux.c pin_mux.h
* BE CAREFUL MODIFYING THIS COMMENT - IT IS YAML SETTINGS FOR THE PINS TOOL **
*/

#include "fsl_common.h"
#include "fsl_port.h"
#include "pin_mux.h"

#define PIN16_IDX          16u  /*!< Pin number for pin 16 in a p
#define PIN17_IDX          17u  /*!< Pin number for pin 17 in a p
#define PIN21_IDX          21u  /*!< Pin number for pin 21 in a p
#define SOPT5_UART0TXSRC_UART_TX  0x00u  /*!< UART 0 transmit data source :

/*
* TEXT BELOW IS USED AS SETTING FOR THE PINS TOOL *****
BOARD_InitPins:
- options: {coreID: singlecore, enableClock: 'true'}
- pin_list:
  - {pin_num: '62', peripheral: UART0, signal: RX, pin_signal: PTB16/SPI1_SOUT/I
  - {pin_num: '63', peripheral: UART0, signal: TX, pin_signal: PTB17/SPI1_SIN/U
  - {pin_num: '67', peripheral: GPIOB, signal: 'GPIO, 21', pin_signal: PTB21/SP
* BE CAREFUL MODIFYING THIS COMMENT - IT IS YAML SETTINGS FOR THE PINS TOOL **
*/

/*FUNCTION*****
*
* Function Name : BOARD_InitPins
* Description   : Configures pin routing and optionally pin electrical feature:
*
*END*****
void BOARD_InitPins(void) {
    CLOCK_EnableClock(kCLOCK_PortB);          /* Port B Clock Ga

    PORT_SetPinMux(PORTB, PIN16_IDX, kPORT_MuxAlt3);          /* PORTB16 (pin 62
    PORT_SetPinMux(PORTB, PIN17_IDX, kPORT_MuxAlt3);          /* PORTB17 (pin 63
    PORT_SetPinMux(PORTB, PIN21_IDX, kPORT_MuxAsGpio);          /* PORTB21 (pin 67
    SIM->SOPT5 = ((SIM->SOPT5 &
        (~(SIM_SOPT5_UART0TXSRC_MASK)))          /* Mask bits to ze
        | SIM_SOPT5_UART0TXSRC(SOPT5_UART0TXSRC_UART_TX)          /* UART 0 transmit
    );
}

/*****
* EOF
*****

```

图 2-9. 生成代码

8. 现在，可将源代码的内容拷贝并粘帖到应用程序和 IDE 中。也可以导出生成的文件。要导出文件：选择“文件 > 导出”（桌面版）或“引脚 > 导出”（网页版）。在“导出”对话框中，选择“导出源文件”选项。

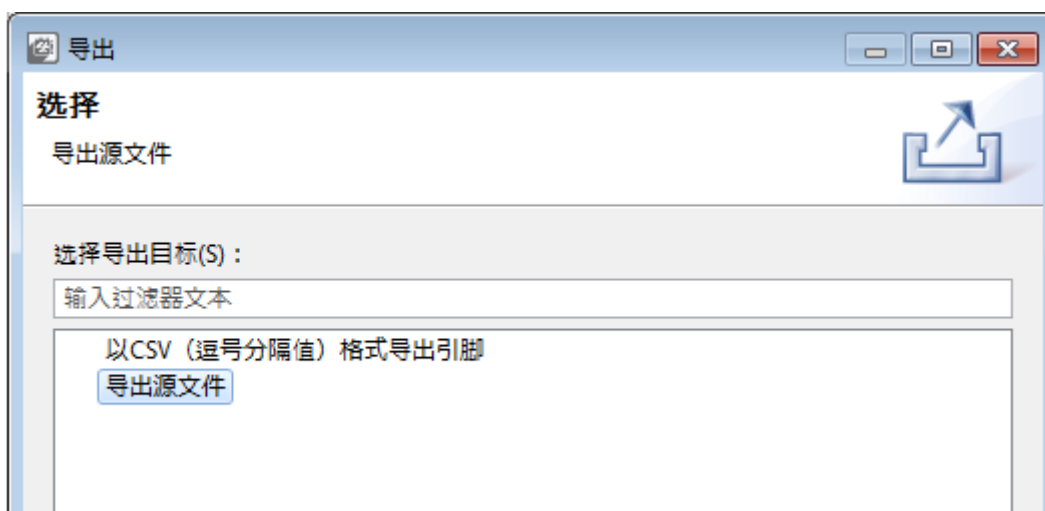


图 2-10. 导出源文件

9. 单击“下一步”，然后指定文件的保存目录。

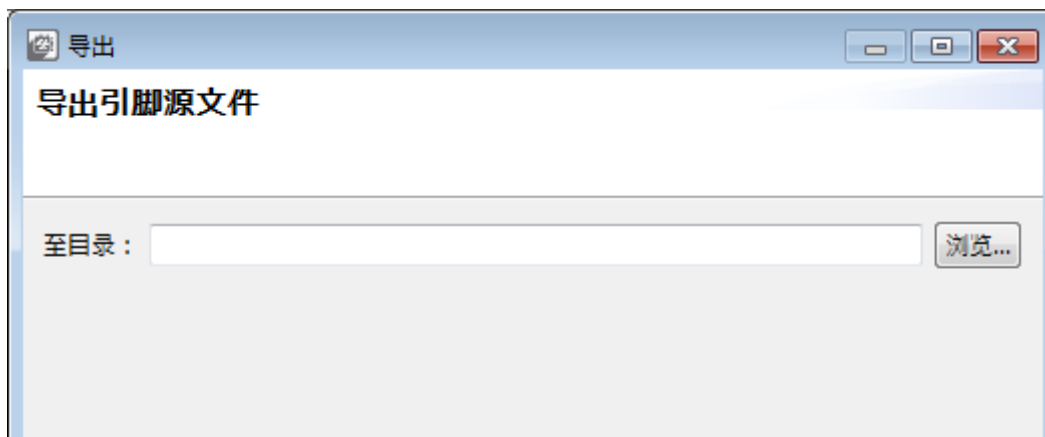


图 2-11. 导出引脚源文件

10. 单击“完成”导出文件。

11. 您可以在应用中整合导出的文件，并将它们用作源文件。

恭喜，您已成功创建了引脚配置！

本用户指南的后续章节将详细介绍该工具的功能。



第 3 章 配置

配置指.mex 文件中存储的常用工具设置。该文件包含所有可用工具的设置，可用于 web 和桌面版本。

3.1 创建新配置

要创建新配置：

1. 如果引脚配置工具已打开，请选择”文件 > 新建“命令。
显示“创建新配置”对话框。
2. 展开产品系列列表并选择所需处理器。

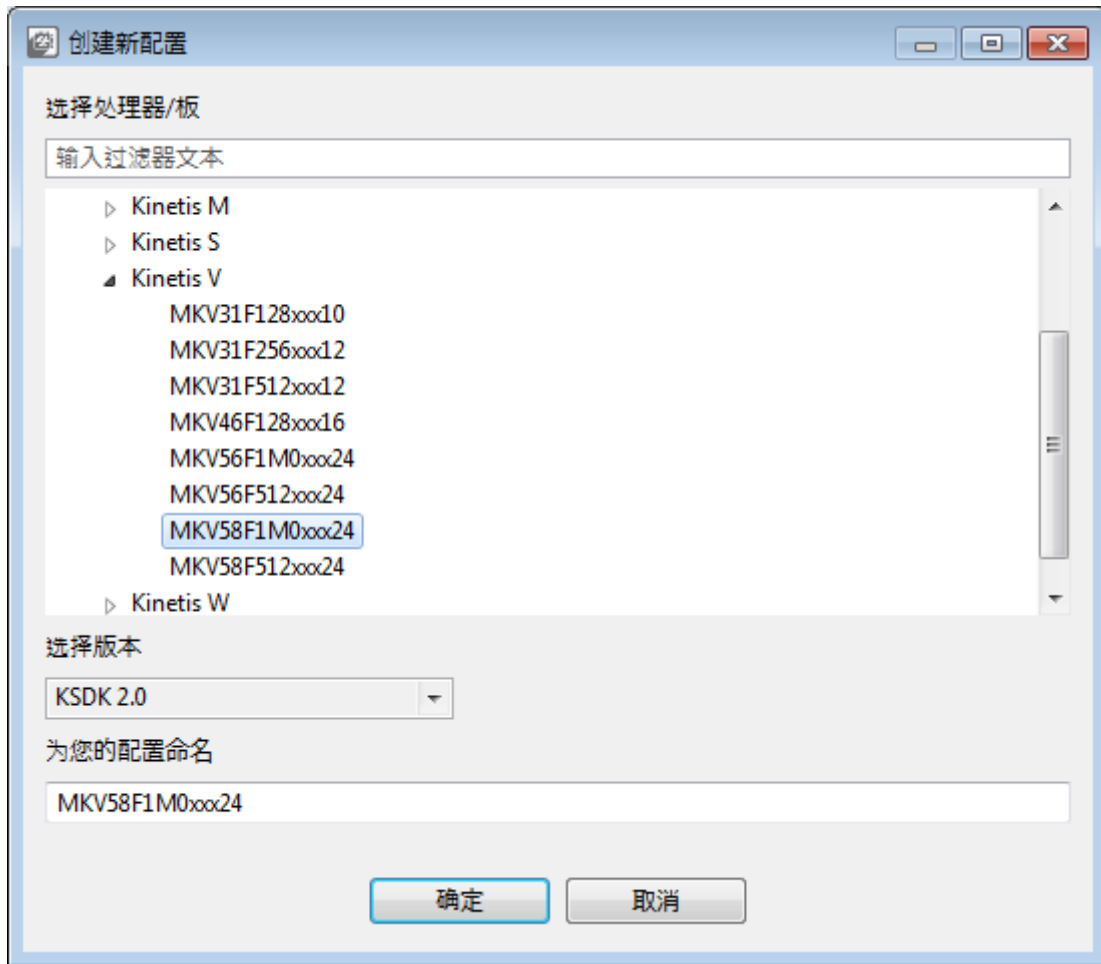


图 3-1. 选择处理器

或者，在“选择处理器/板”文本框中输入处理器名称，然后从筛选的结果中选择处理器。

3. 如需为配置指定不同名称，编辑“为您的配置命名”对话框中的文本。
4. 单击“确定”按钮。

选择的配置便会出现在引脚配置工具中，您现在可以进行配置。

5. 您现在可以添加您需要配置的外设、封装和引脚，并为选定的引脚分配/配置引脚复用。

注

您可以通过网页顶部的下拉菜单创建新配置。

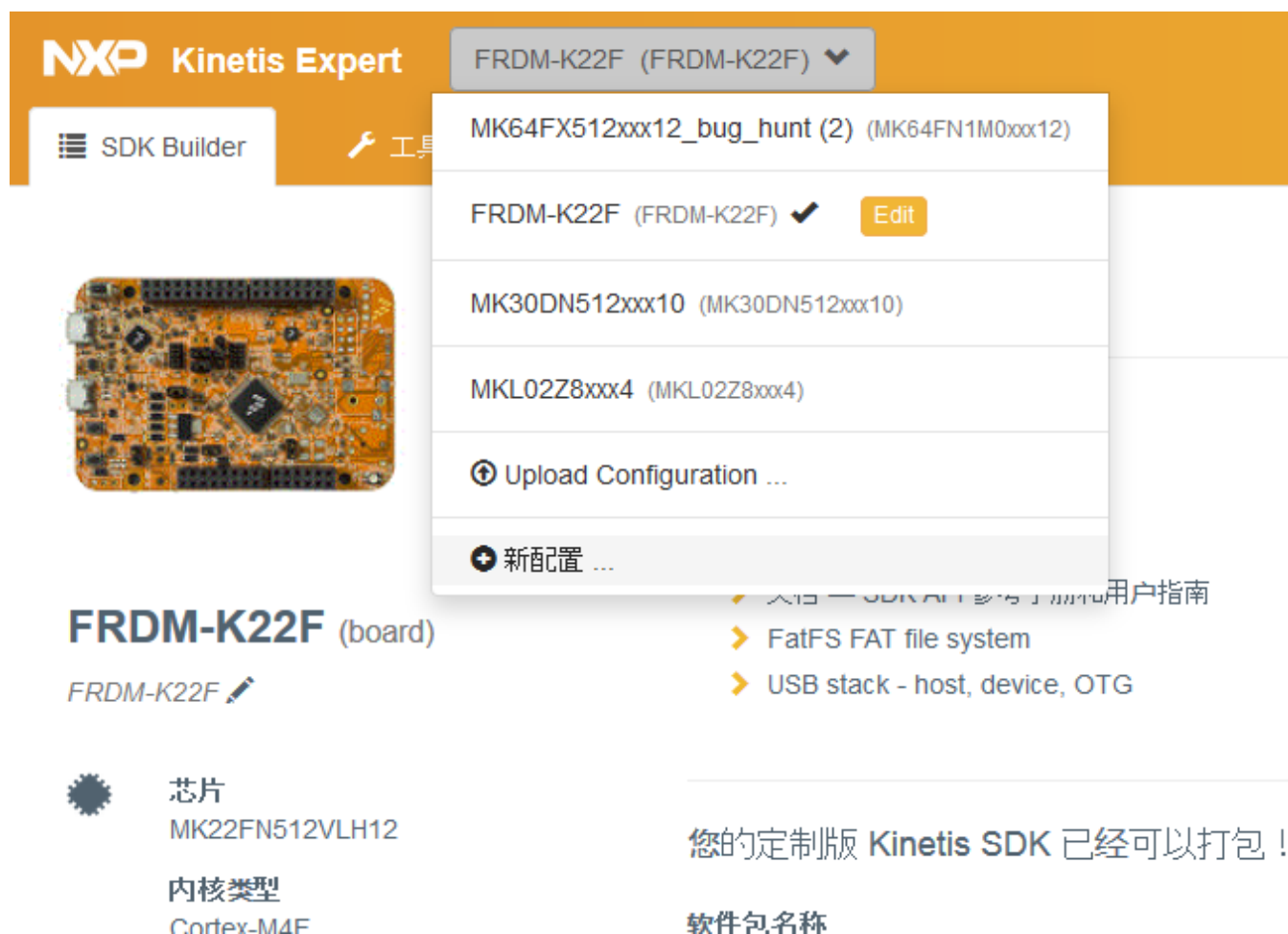


图 3-2. 网页上的下拉菜单

3.2 保存配置

如需保存配置或配置文件，请选择“文件 > 另存为”。

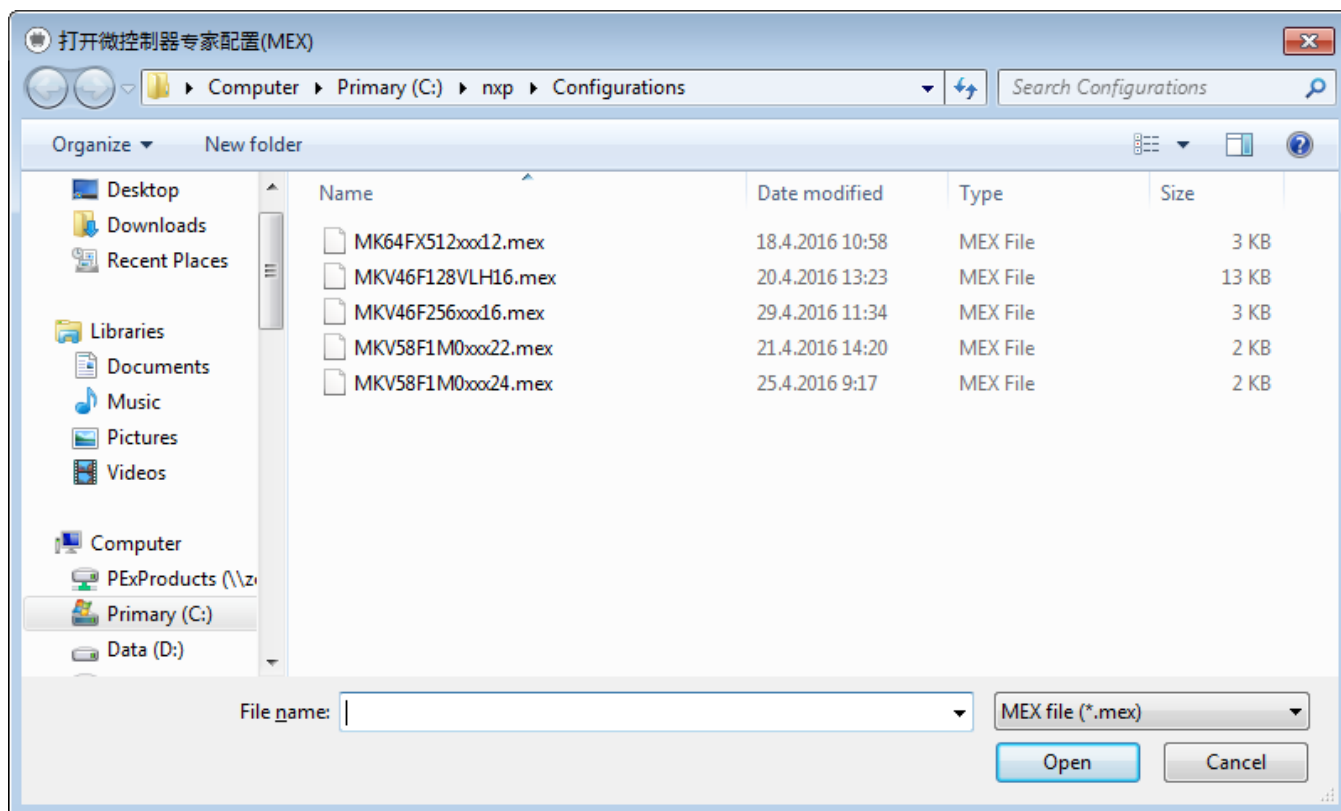


图 3-3. 保存配置

注

在桌面版中，配置会采用.mex 文件扩展名保存。

3.3 打开现有配置

如需打开之前保存的配置文件：

1. 请选择“文件 > 打开”。

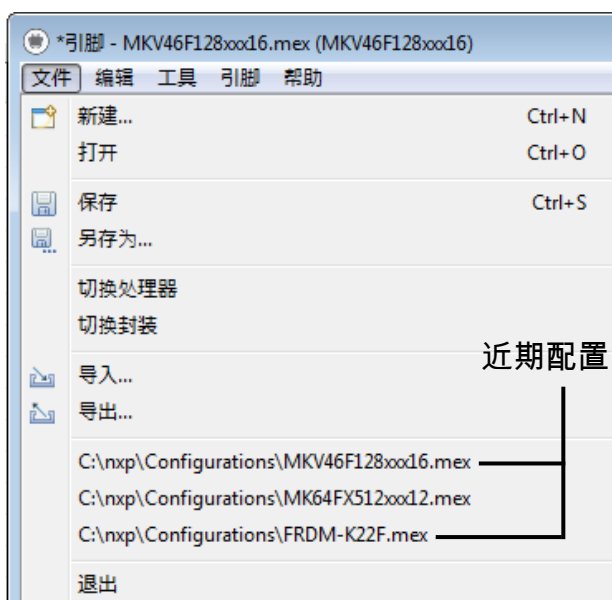


图 3-4. 打开配置文件

出现打开配置文件对话框。

2. 导航到保存之前配置文件的文件夹。

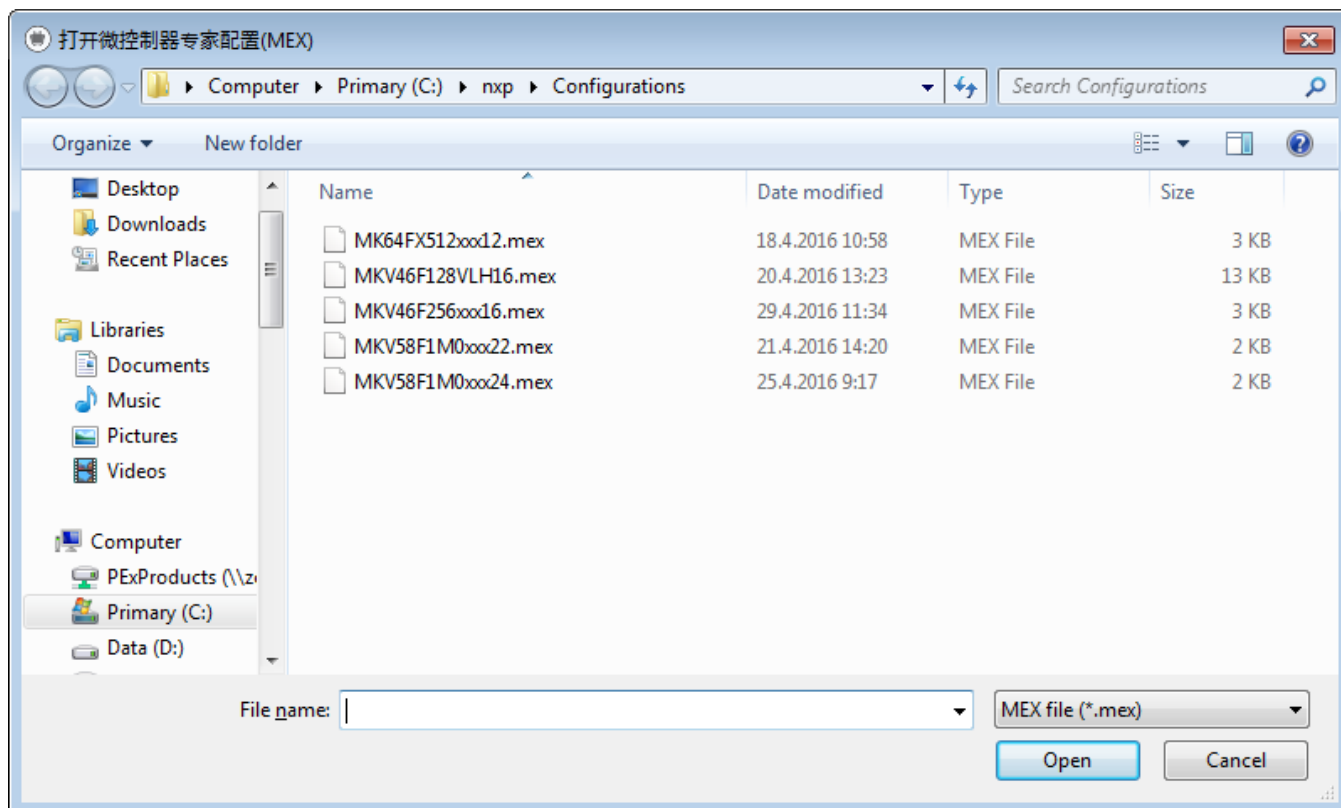


图 3-5. 打开对话框

3. 选择配置文件然后单击“打开”。

现有配置文件出现在引脚配置工具中。

第 4 章 引脚配置工具

本章介绍引脚配置工具的原理及其用于生成引脚路由和多路复用的方法。

4.1 选择引脚配置工具

通过菜单或桌面版上安装的快捷方式来选择引脚配置工具。

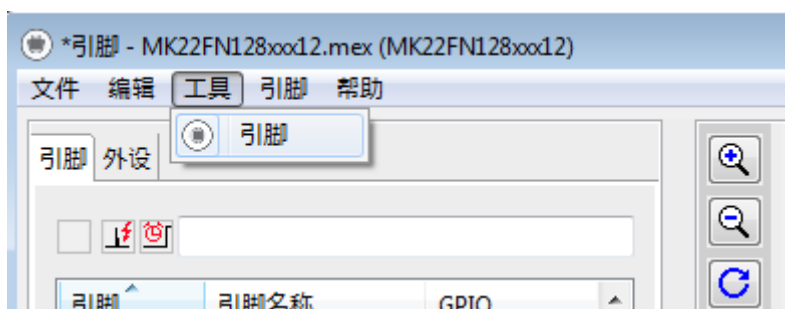


图 4-1. 选择引脚配置工具 - 桌面版



图 4-2. 选择引脚配置工具 - 网页版

4.2 引脚路由原则

引脚配置工具用于将外设信号配置到引脚或内部信号。

如需定义路由路径，请选择：

1. 外设
2. 信号
3. 引脚

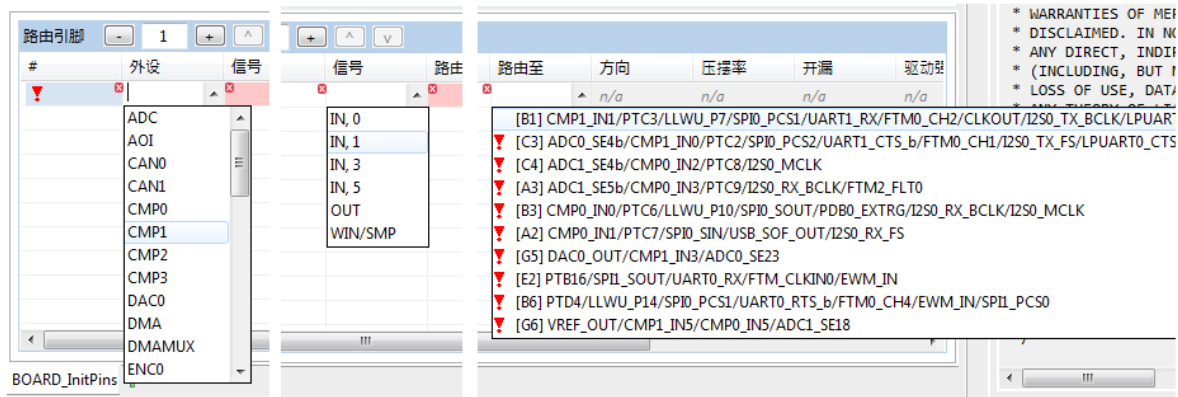


图 4-3. 定义路由路径

根据选定的“外设”，选择一个可用信号。

将选定的“信号”路由至所需引脚。

选择一个非冲突/可用引脚。选择“外设”、“信号”、以及“路由至”之后，引脚配置即已完成。也可稍后配置引脚电气功能。

4.3 用户界面

引脚配置工具包含几个视图。

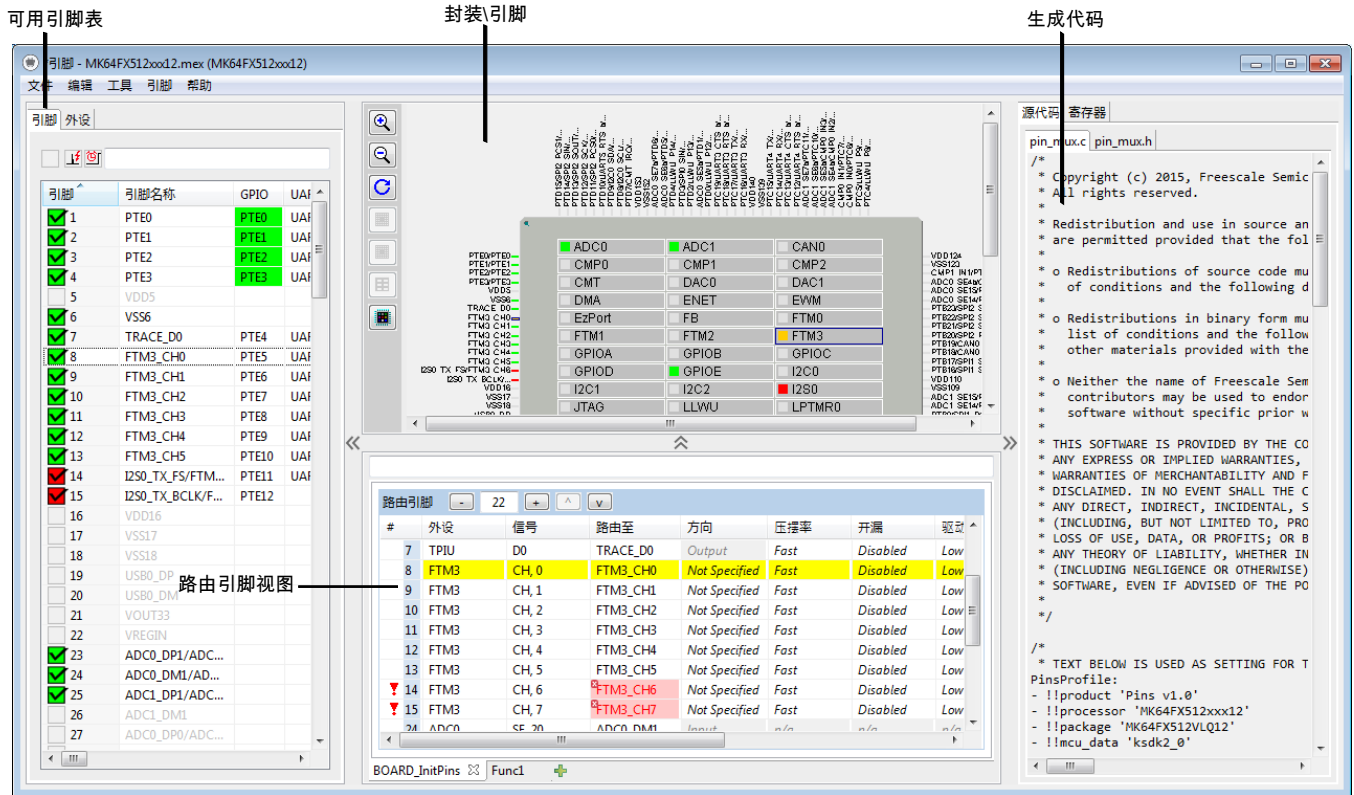


图 4-4. 引脚配置工具用户界面

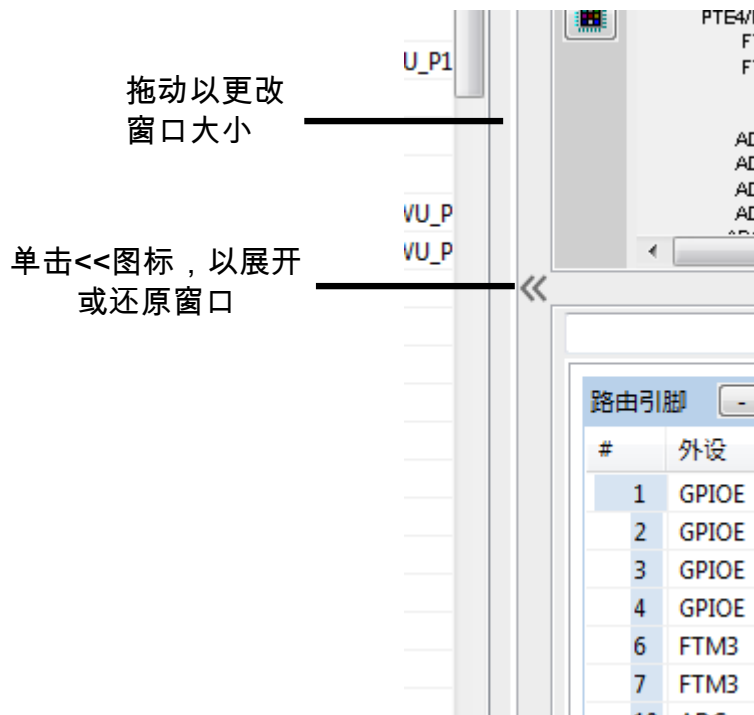


图 4-5. 调整/恢复窗口大小

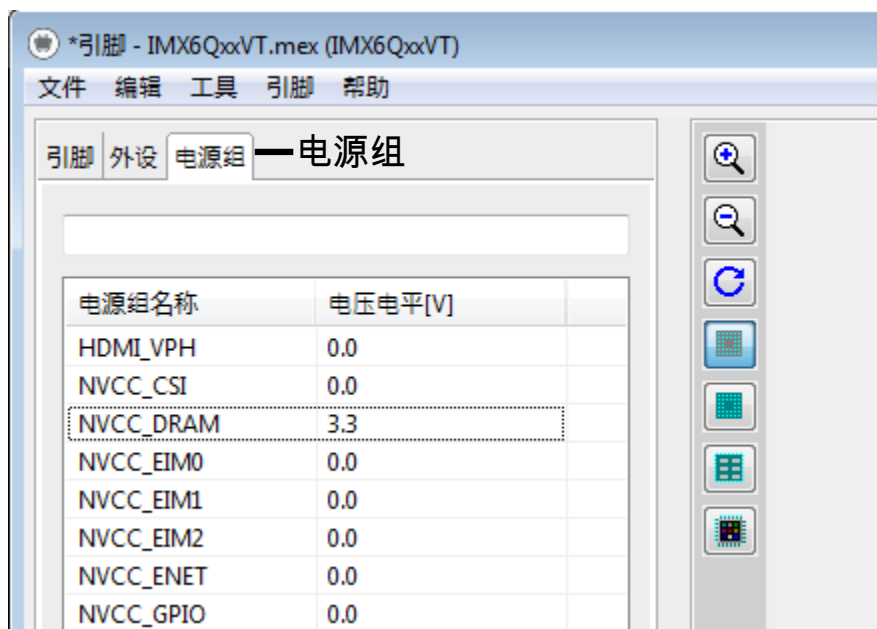


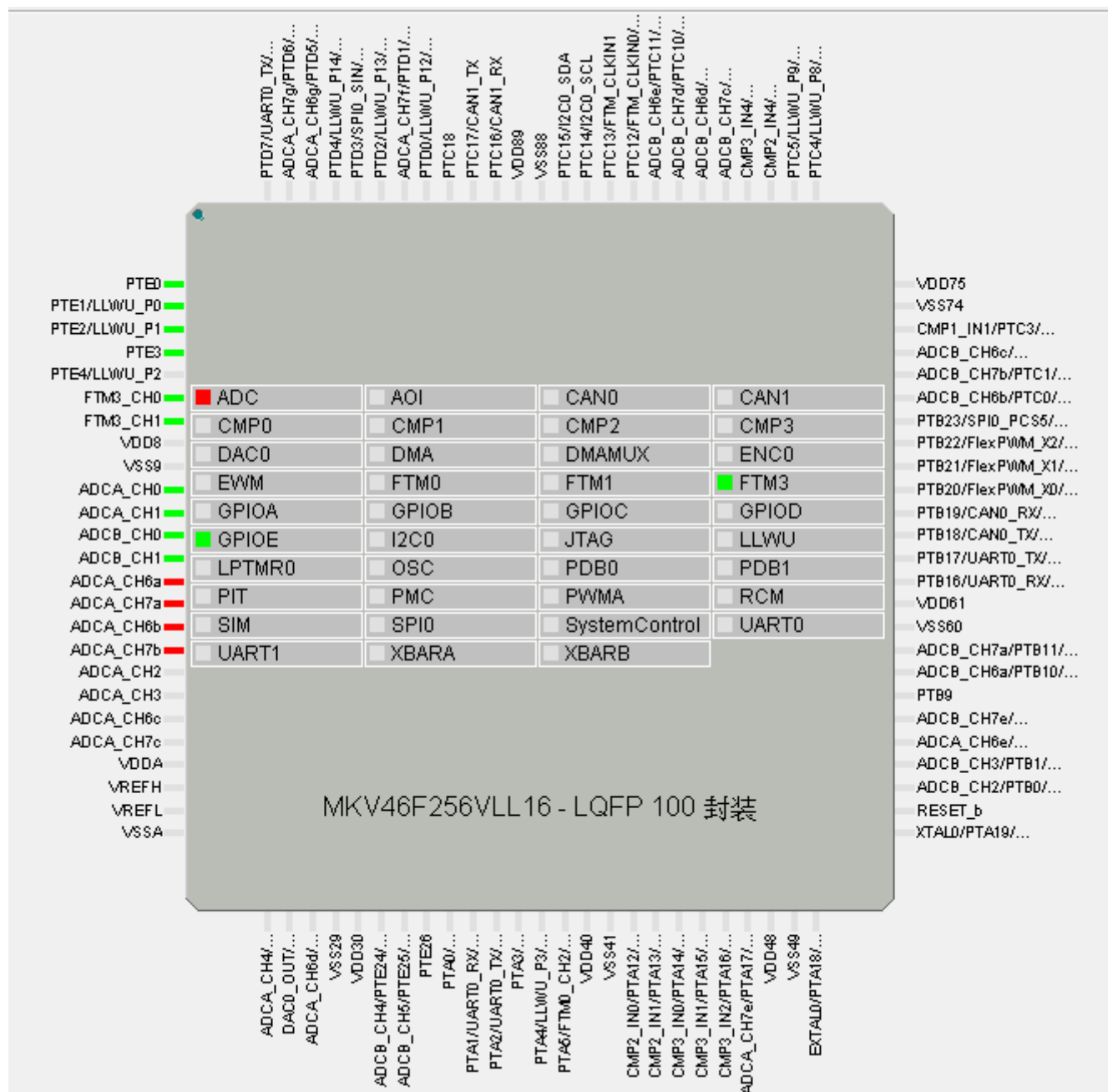
图 4-6. 选择功能组

注

并非所有处理器都支持功能组。

4.3.1 封装

处理器封装显示在引脚配置工具窗口的中间。处理器封装显示包括资源分配在内的封装整体概况。



该视图显示带引脚位置的封装概况。中间是外设。

使用“显示外设”图标即可查看 BGA 封装。








绿色表示已经路由的引脚/外设。

灰色表示引脚/外设尚未路由。

该视图还显示封装变量和描述（类型和引脚数量）。

工具栏中有下列图标：

表 4-1. 工具栏选项

图标	说明
	放大封装图像。
	缩小封装图像。
	旋转封装图像。
	显示引脚，以便从底部查看。仅 BGA 封装提供此选项。
	显示引脚，以便从顶部查看。仅 BGA 封装提供此选项。
	显示外设。仅 BGA 封装提供此选项。
	切换封装。

注

并非所有视图都可用，具体取决于选定的处理器封装。

单击“切换封装”图标便会显示“为处理器选择封装”对话框。



图 4-7. 切换封装

“为处理器选择封装”对话框会显示可用处理器封装列表，其中包含封装类型和引脚数量信息。

4.3.2 路由引脚视图

该路由引脚视图显示了路由引脚的列表，并协助进行引脚配置。该视图还可以配置引脚的电气属性并显示所有引脚。视图显示配置中的可用面板配置，其中各引脚与信号名称和功能关联。

注

该电气功能配置仅适用于表格中的引脚，例如路由引脚。

创建新项目时，该表格为空，也就是说未配置任何引脚。每行代表一个引脚的配置，如果没有冲突，代码会立即更新。

使用表格下拉菜单来配置引脚。要配置引脚，请按从左往右的顺序，首先选择外设，然后选择所需信号，最后选择路由引脚。

查看表格右边部分以配置电气特性。




如果不支持该功能，则会显示 n/a。

#	外设	信号	路由至	方向	压摆率	开漏	驱动强度	拉动选择
1	GPIOE	port, 0	PTE0	Output	Fast	Enabled	Low	Pulldown
2	GPIOE	port, 1	PTE1/LLWU_P0	<i>Not Specified</i>	Fast	Disabled	Low	Pulldown
3	GPIOE	port, 2	PTE2/LLWU_P1	<i>Not Specified</i>	Fast	Disabled	Low	Pulldown
4	GPIOE	port, 3	PTE3	<i>Not Specified</i>	Fast	Disabled	Low	Pulldown
6	FTM3	tmr_ch, 0	FTM3_CH0	<i>Not Specified</i>	Fast	Disabled	Low	Pulldown
7	FTM3	tmr_ch, 1	FTM3_CH1	<i>Not Specified</i>	Fast	Disabled	Low	Pulldown

图 4-8. 路由引脚视图

灰色背景表示为只读项目。

斜体值表示非配置值，为复位后的值，且没有生成代码，因此配置由复位后的值决定。

	用斜体显示复位后的值。若在其他功能中配置，实际值可能与复位后的值有所不同。
使用下拉菜单选择所需值。	
	如果您选择与复位后的值相似的值，则工具将会始终生成代码以置位该功能。
使用下拉菜单来将值复位至复位后状态。	
	若项目不支持复位至复位后的值，或当前值已能够代表复位后的状态，则 Reset 菜单不可用。

第一行显示 BGA 封装的引脚编号或坐标。

4.3.3 视图控制

下图显示路由引脚的视图控制。

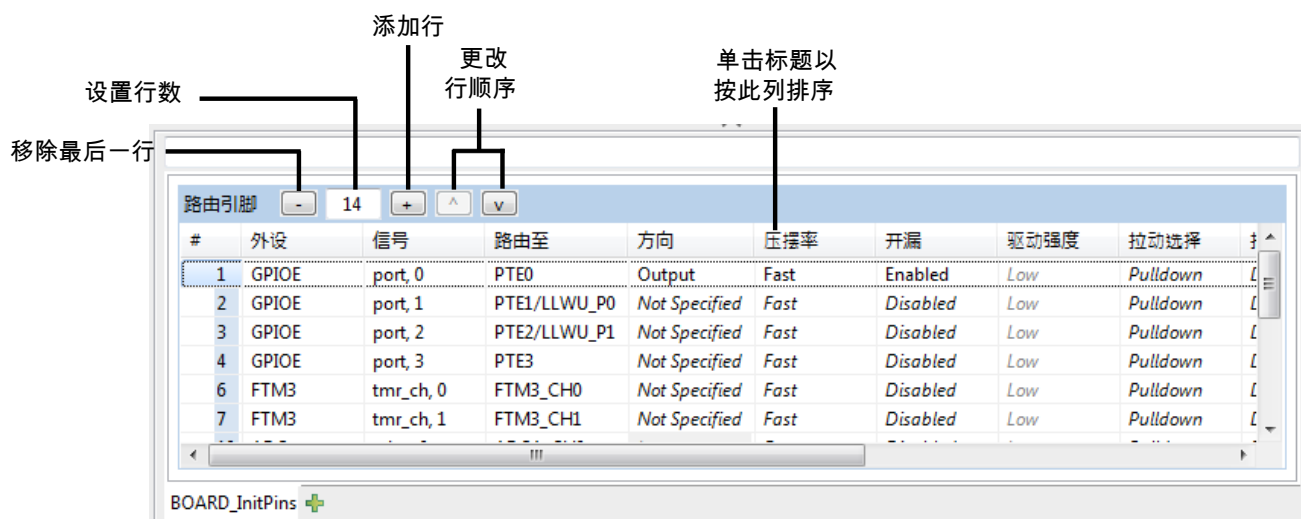


图 4-9. 视图控制

添加/移除行:

- 如需在表格尾部添加新行，请单击[+]图标。
- 如需移除最后一行，请单击[-]图标。
- 如需在指定位置删除指定行或插入新行，右击使用弹出菜单命令。

添加特定数目行或清除表格:

- 如需添加特定数目行，请指定确切的行数。
- 如需清除表格，输入 0。

更改行顺序:

如需更改行顺序，使用箭头图标将行向上或向下移动。

4.3.4 筛选路由引脚

下图显示了路由引脚视图的筛选区域。

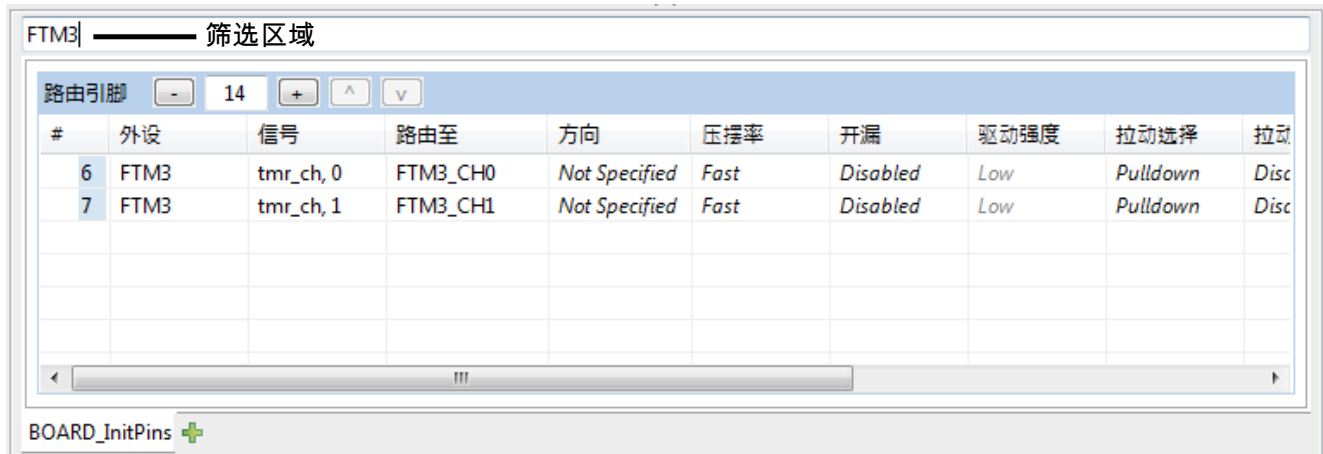


图 4-10. 筛选区域

如需即时筛选行，在筛选区域中输入文本或搜索短语。

注

输入搜索词之后，它也会在引脚完整名称中搜索，并显示包含搜索词的行。

4.3.5 高亮显示和颜色代码

高亮显示可以轻易识别封装上的路由引脚/外设。默认情况下，当前选择（引脚/外设）在封装视图中高亮显示。

- 错误（红色）表示引脚有错误。
- 已使用（绿色）表示引脚为多路复用。
- 未使用（浅灰色）表示引脚可用于多路复用，但尚未使用。
- 选定（引脚边框为黄色，保持其他颜色可见）表示引脚已被选定。例如按外设或按引脚。

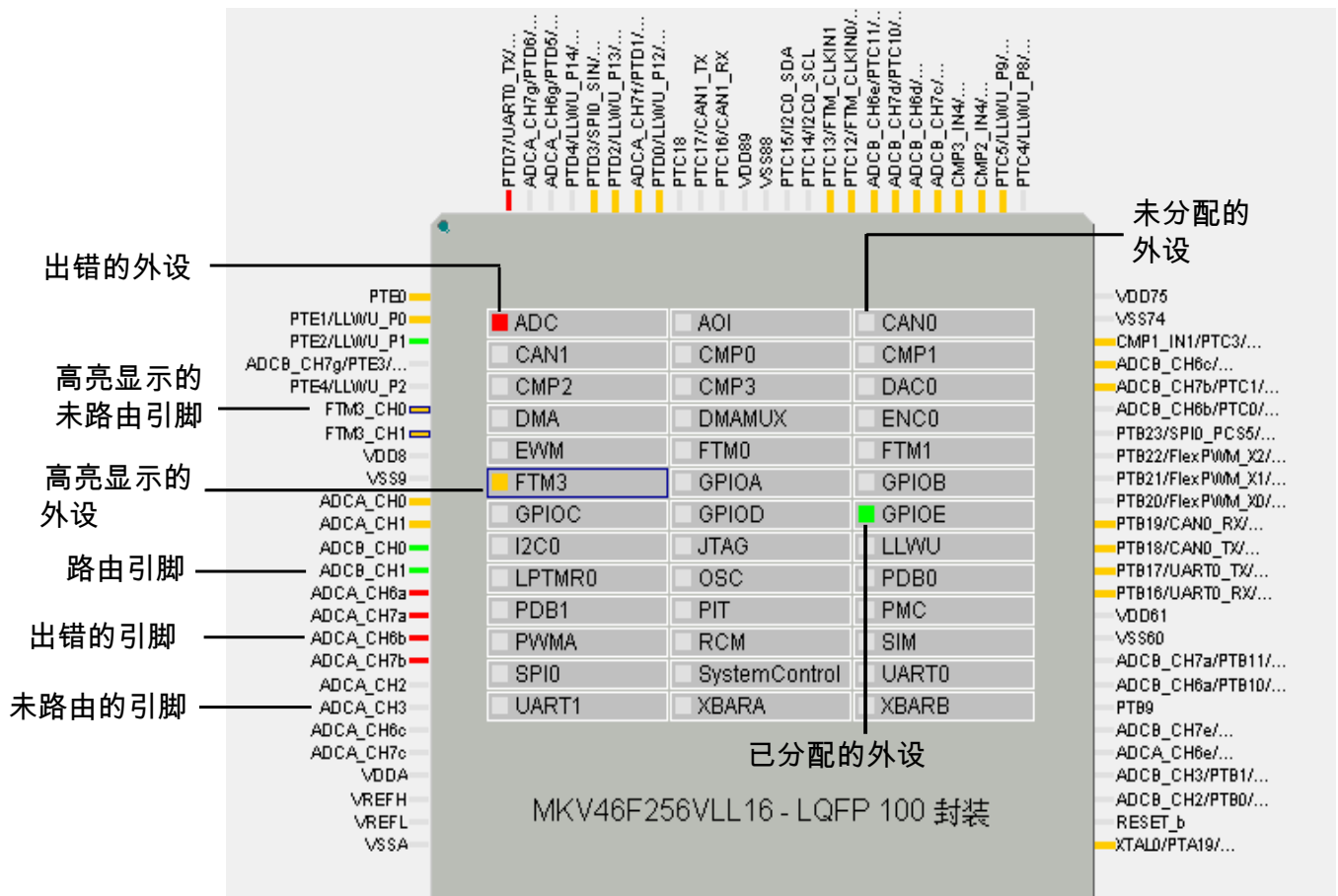


图 4-11. 高亮显示和颜色代码

#	外设	信号	路由至	方向	压摆率	开漏
4	GPIOE	GPIO_3	PTE3	Not Specified	Fast	Disable
6	SPI1	PCS2	SPI1_PCS2	Output	Fast	Disable
7	GPIOE	GPIO_6	PTE6	Not Specified	Fast	Disable
7	SPI1	PCS3	SPI1_PCS3	Output	Fast	Disable
71			ADC0_SE15/PTC1/LLWU_P6/SPI0_PCS3/UART1_RTS_b/FTM0_CH0/I2S0_TXD0/LPUART0_RTS_b	n/a	Fast	Disable
				n/a	n/a	n/a

BOARD_InitPins

图 4-12. 引脚冲突

A2	PMU	pcie_rext	PCIE_REXT		Input
H20	I2C1	scl	EIM_DATA...	NVCC_EIM...	Input/Out...
N6	I2C1	sda	CSI0_DAT...	NVCC_CSI...	Input/Out...

图 4-13. 警告

- 封装视图
 - 单击外设或使用弹出菜单以高亮显示外设:

- 以及所有分配引脚（选定外设）
- 或所有可用引脚（如尚未分配）
- 单击引脚或使用弹出菜单以高亮显示引脚和外设。
- 单击封装外部以取消高亮显示。
- 外设/引脚视图
 - 外设和引脚操作如上图所述
 - 键盘可用于进行（多项）选择。这仅使用于桌面版。

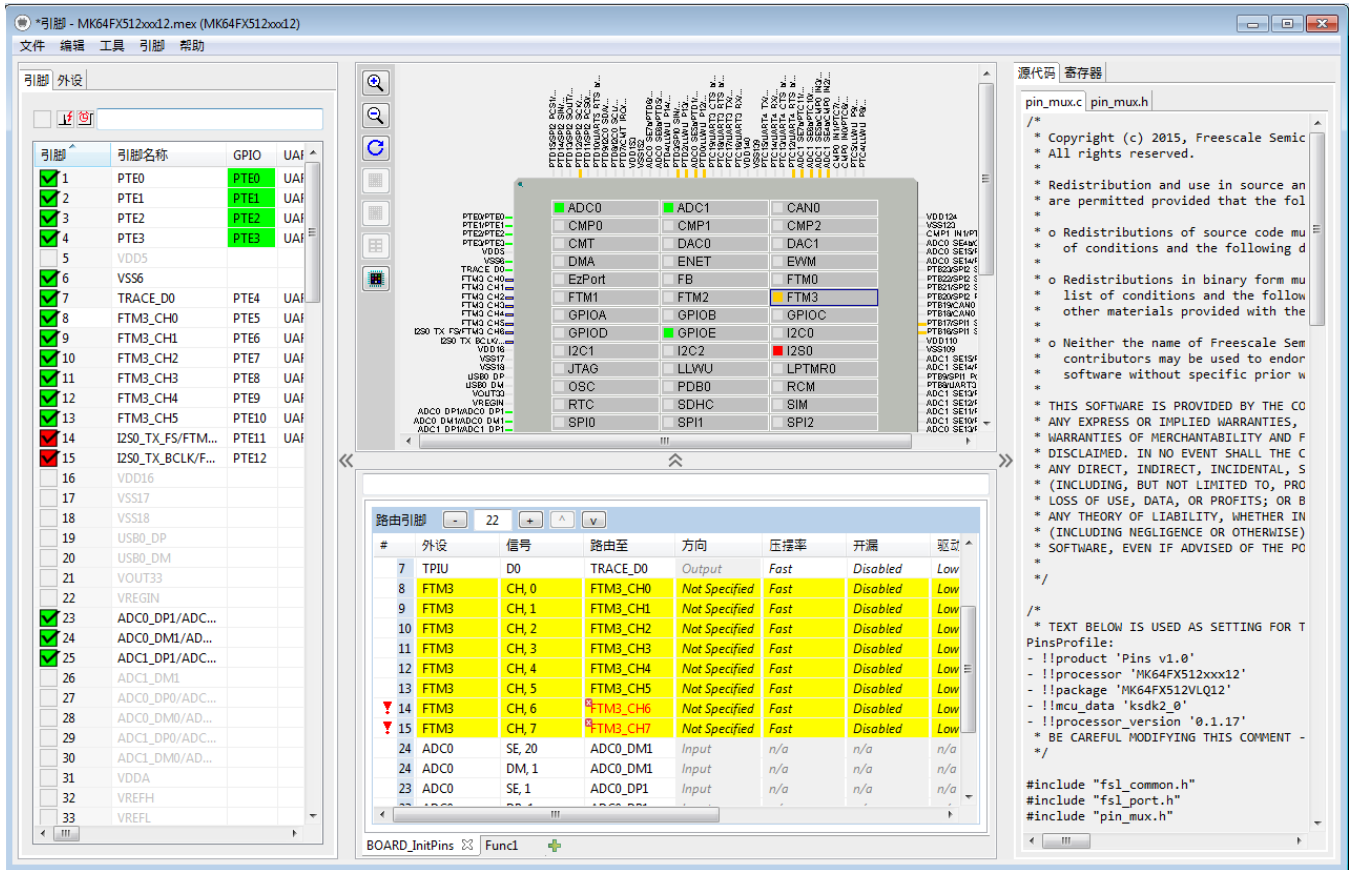



图 4-14. 引脚配置工具界面

 桌面版工具可以在引脚下拉菜单遍历引脚时高亮显示引脚。

4.3.6 在引脚视图中筛选

下图显示了引脚视图中的筛选控制。

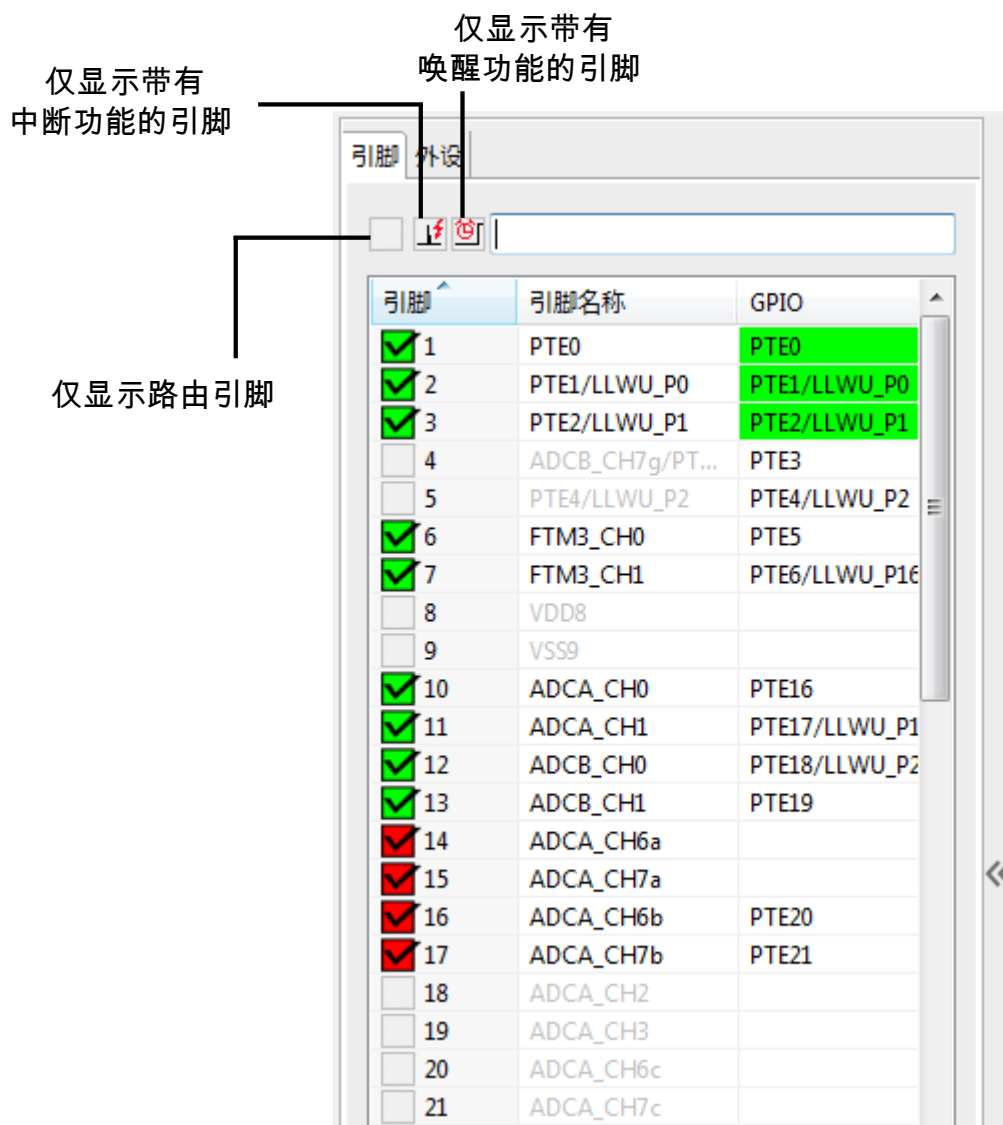


图 4-15. 筛选控制

输入任意文本搜索整个表格。该操作将会搜索引脚功能，如唤醒或中断功能。

若要仅查看选定外设上的引脚，单击标题按名称排列指定外设的引脚

单击第一个复选框图标，查看全部引脚或仅查看路由引脚。

4.3.7 函数

“函数”用于对一组路由引脚进行分类，另外还会以函数形式为该配置创建代码，以便之后可由应用调用。

工具可以创建出多个可用于配置引脚多路复用的函数。

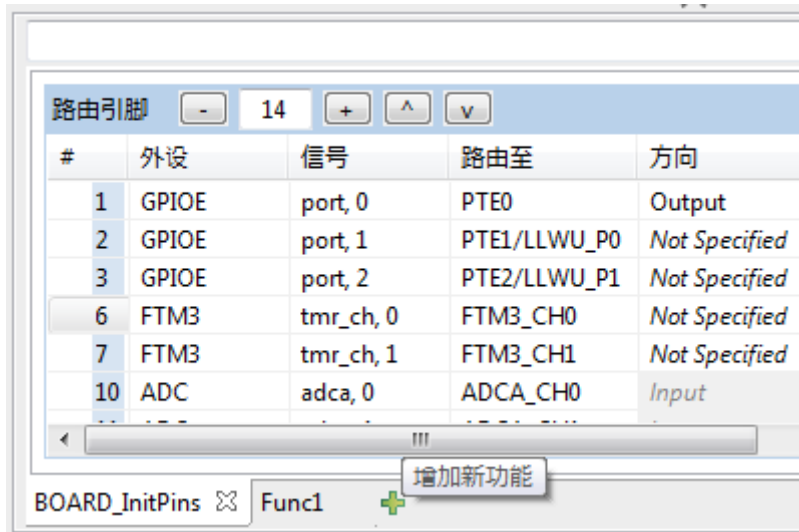


图 4-16. 路由引脚

工具既不会限制每个函数使用引脚，也不会显示每个函数的引脚使用状况。每个函数可以定义一组路由引脚或重新配置已路由的引脚。

配置中指定多个函数时，封装视图主要显示选定函数的引脚和外设。不同函数的引脚和外设以浅透明度显示且无法配置，直至切换至该函数。

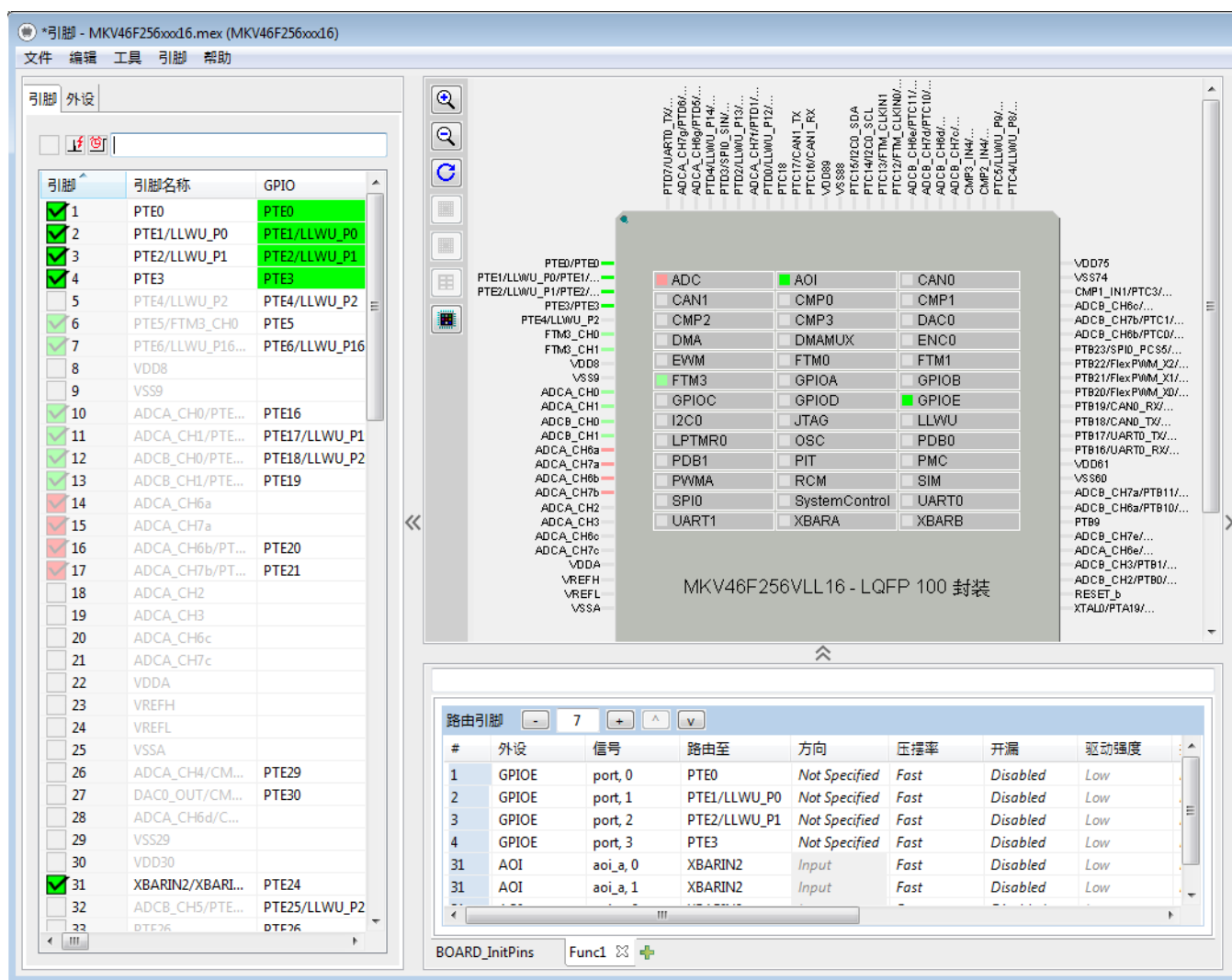


图 4-17. 引脚和外设

4.3.8 外设视图

“外设”视图显示外设及其信号列表。仅“外设”和“引脚”视图会显示状态复选框（已分配）。

表 4-2. 状态代码

颜色代码	状态
<input checked="" type="checkbox"/> port, 1	错误
<input checked="" type="checkbox"/> port, 2	已分配
<input type="checkbox"/> port, 3	可用
<input checked="" type="checkbox"/> port, 16	警告

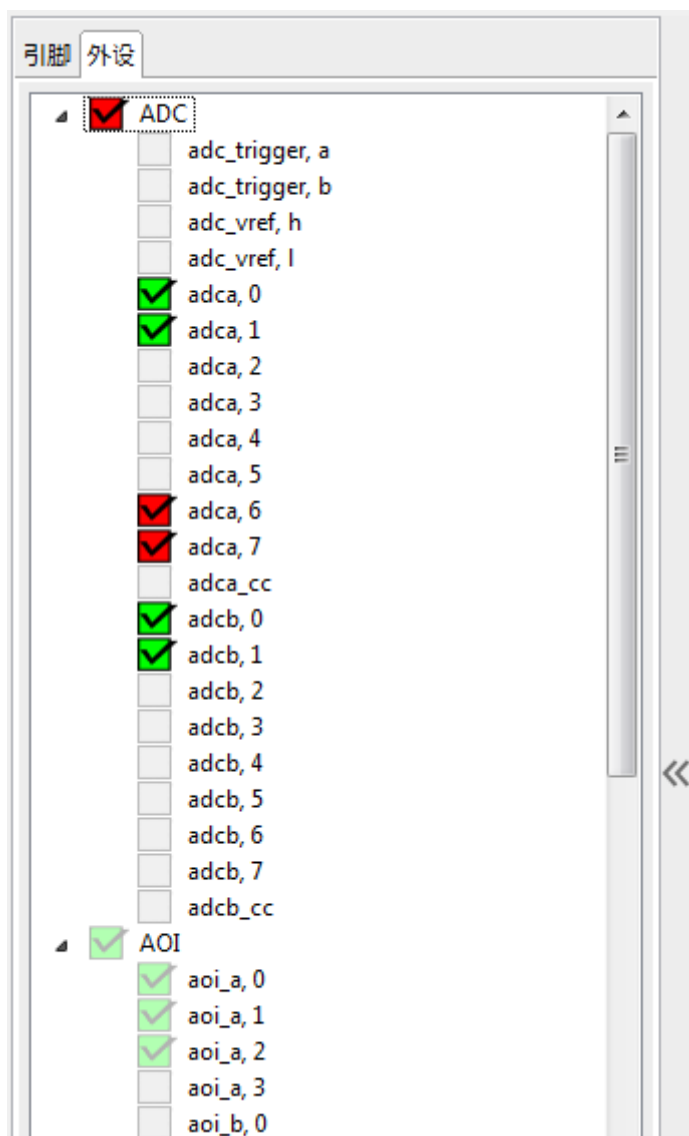


图 4-18. 外设选项卡

使用复选框来路由/不路由选定引脚。

如需路由/不路由多个引脚，请单击外设并从选择信号对话框中选择选项。

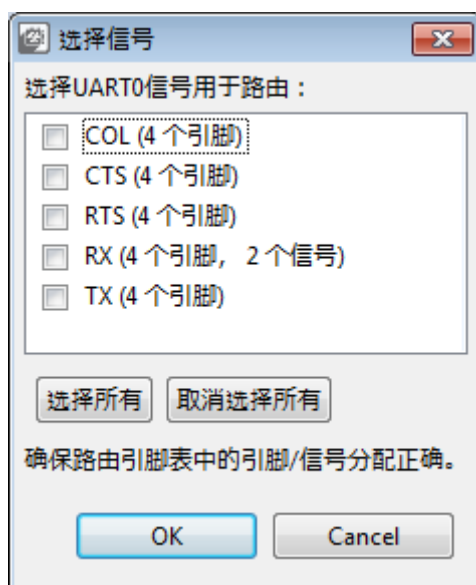


图 4-19. 选择信号对话框



如果多个信号可路由至一个引脚，则会显示[...]。单击后便会出现多个信号配置对话框。

4.3.9 引脚表

”引脚表”视图以表格形式显示所有引脚。

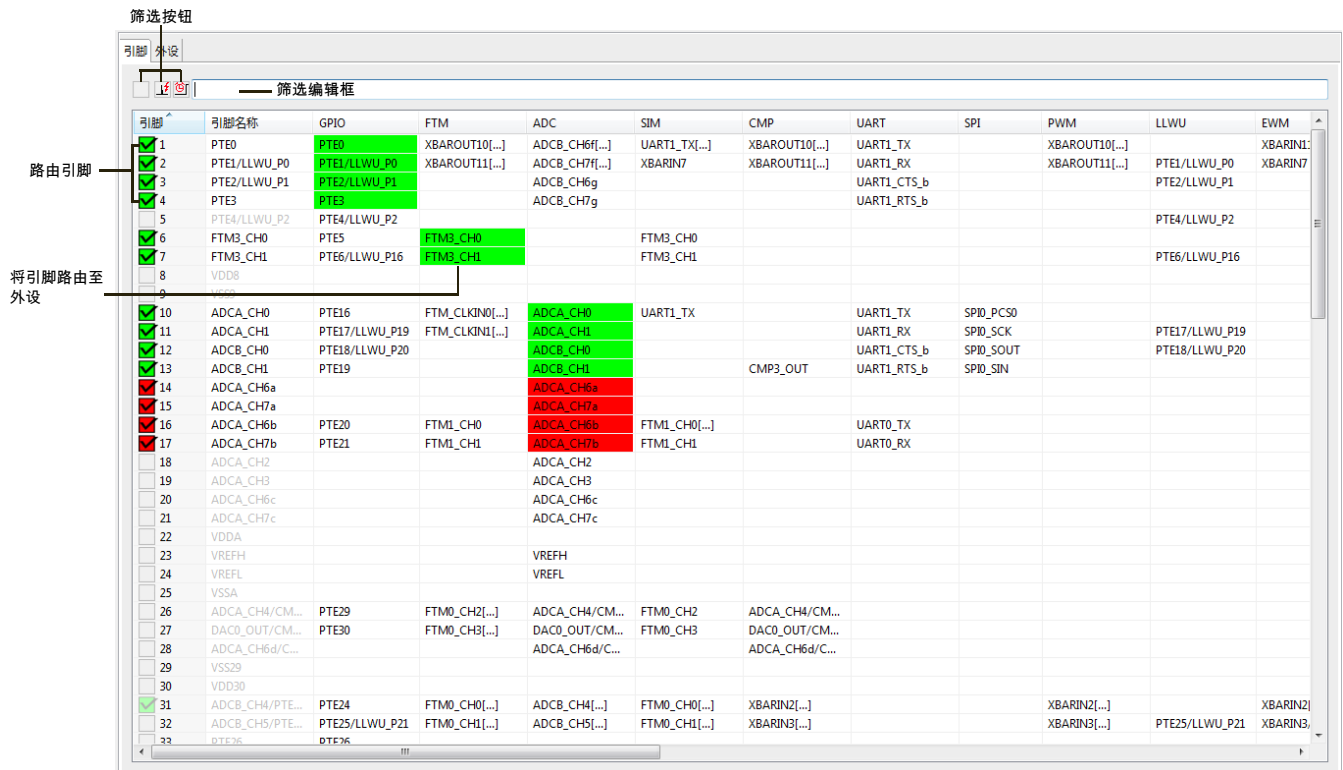


图 4-20. 引脚表视图

该视图显示指定器件上的所有可用引脚。“引脚名称”列显示引脚的默认名称或引脚的路由状态。引脚名称还会更改为显示选定外设（如果路由）的适当函数。表格下一列显示指定外设的外设和引脚名称。若外设项目较少，则会在最后一列累积。

如需路由/不需路由引脚至指定外设，单击表格的单元格即可。路由引脚勾选复选框并显示为绿色。有颜色的单元格即表示引脚已路由至指定外设。若路由出现冲突，则会显示红色。

单击指定单元格或勾选第一列的复选框即可取消路由。

每个路由引脚都会出现在路由引脚表格中。

配置中指定多个函数时，引脚表格视图主要显示选定函数的引脚。不同函数的引脚以浅透明度显示且无法配置，直至切换至该函数。

引脚	引脚名称	GPIO	FTM	ADC	SIM	CMP
<input checked="" type="checkbox"/>	1	PTE0	XBAROUT10[...]	ADCB_CH6f[...]	UART1_TX[...]	XBAROUT10[...]
<input checked="" type="checkbox"/>	2	PTE1/LLWU_P0	XBAROUT11[...]	ADCB_CH7f[...]	XBARIN7	XBAROUT11[...]
<input checked="" type="checkbox"/>	3	PTE2/LLWU_P1		ADCB_CH6g		
<input checked="" type="checkbox"/>	4	PTE3		ADCB_CH7g		
<input type="checkbox"/>	5	PTE4/LLWU_P2				
<input checked="" type="checkbox"/>	6	PTE5/FTM3_CH0	FTM3_CH0		FTM3_CH0	
<input checked="" type="checkbox"/>	7	PTE6/LLWU_P16...	FTM3_CH1		FTM3_CH1	
<input type="checkbox"/>	8	VDD8				
<input type="checkbox"/>	9	VSS9				
<input checked="" type="checkbox"/>	10	PTE16	FTM_CLKIN0[...]	ADCA_CH0	UART1_TX	
<input checked="" type="checkbox"/>	11	PTE17/LLWU_P19	FTM_CLKIN1[...]	ADCA_CH1		
<input checked="" type="checkbox"/>	12	PTE18/LLWU_P20		ADCB_CH0		
<input checked="" type="checkbox"/>	13	PTE19		ADCB_CH1		CMP3_OUT
<input checked="" type="checkbox"/>	14			ADCA_CH6a		
<input checked="" type="checkbox"/>	15			ADCA_CH7a		
<input checked="" type="checkbox"/>	16	PTE20	FTM1_CH0	ADCA_CH6b	FTM1_CH0[...]	
<input checked="" type="checkbox"/>	17	PTE21	FTM1_CH1	ADCA_CH7b	FTM1_CH1	
<input type="checkbox"/>	18			ADCA_CH2		
<input type="checkbox"/>	19			ADCA_CH3		
<input type="checkbox"/>	20			ADCA_CH6c		
<input type="checkbox"/>	21			ADCA_CH7c		

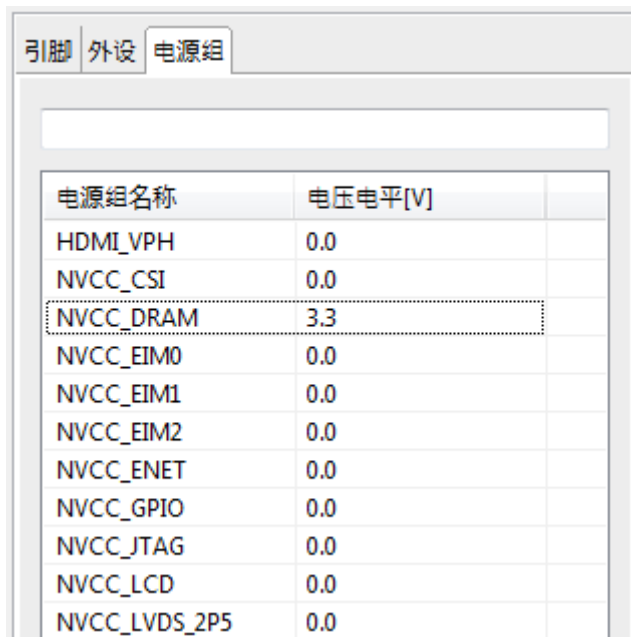
图 4-21. 路由多个函数示例

4.3.10 电源组

如果处理器支持电源组，则“引脚”和“外设”旁会额外显示一个选项卡，以便对其进行配置。

注

并非所有器件都支持这个功能。



电源组名称	电压电平[V]	
HDMI_VPH	0.0	
NVCC_CSI	0.0	
NVCC_DRAM	3.3	
NVCC_EIM0	0.0	
NVCC_EIM1	0.0	
NVCC_EIM2	0.0	
NVCC_ENET	0.0	
NVCC_GPIO	0.0	
NVCC_JTAG	0.0	
NVCC_LCD	0.0	
NVCC_LVDS_2P5	0.0	

图 4-22. 选择电源组

4.3.11 寄存器视图

寄存器视图显示指定外设的寄存器列表。该视图显示寄存器名称、路由配置使用的值以及复位后的值。该值通过应用程序初始化期间生成的代码写入寄存器或位。这是路由函数写入寄存器的最后一个值。复位列包含复位后寄存器中的默认值。寄存器值以十六进制和二进制形式显示。若寄存器（或位）的值未定义，则不会显示值，而是显示问号“?”。

寄存器名称	设置值	复位值	
▲ 外设寄存器			
▶ PORTA_PCR0	0x00000742	0x00000742	寄存器
▲ PORTA_PCR1	0x00000143	0x00000743	
reserved	0b00000000	0b00000000	近期更改的寄存器 以黄色高亮显示
ISF	0b0	0b0	
reserved	0b0000	0b0000	
IRQC	0b0000	0b0000	
LK	0b0	0b0	寄存器的设置值
reserved	0b0000	0b0000	
MUX	0b001	0b111	寄存器复位后的值
reserved	0b0	0b0	
DSE	0b1	0b1	
ODE	0b0	0b0	
PFE	0b0	0b0	
reserved	0b0	0b0	寄存器详情
SRE	0b0	0b0	
PE	0b1	0b1	
PS	0b1	0b1	
▶ PORTA_PCR2	0x00000743	0x00000743	
▶ PORTA_PCR3	0x00000743	0x00000743	
▶ PORTA_PCR4	0x00000743	0x00000743	
▶ PORTA_PCR5	0x00000041	0x00000041	

图 4-23. 寄存器视图

4.4 错误和警告

引脚配置工具检查路由冲突以及配置错误。仅针对选定函数检查路由冲突。可以在不同函数中配置同一个引脚的不同路由，以重新配置动态引脚路由。

#	外设	信号	路由至	方向	压摆率	开/关
4	GPIOE	GPIO, 3	PTE3	Not Specified	Fast	Dis
6	SPI1	PCS2	SPI1_PCS2	Output	Fast	Dis
7	GPIOE	GPIO, 6	PTE6	Not Specified	Fast	Dis
7	SPI1	PCS3	SPI1_PCS3	Output	Fast	Dis

图 4-24. 错误和警告

出现错误时，行第一列会显示冲突，出现冲突的单元格中会显示错误。上图显示了发生配置错误的外设/信号。详细错误消息会显示为工具提示。

4.4.1 未完成的路由

未完成路由的单元格以红色背景显示。如需生成适当的引脚路由，单击下拉箭头并选择合适值。单元格中的红色标记表示错误情况。

#	外设	信号	路由至	方向	压摆率	开/关
4	GPIOE	GPIO, 3	PTE3	Not Specified	Fast	Dis
6	SPI1	PCS2	SPI1_PCS2	Output	Fast	Dis
7	GPIOE	GPIO, 6	PTE6	Not Specified	Fast	Dis
7	SPI1			n/a	Fast	Dis

Name of the selected pin/signal function. . .

ERROR: Peripheral signal未选择

图 4-25. 未完成的路由

单元格工具提示显示冲突或错误的更多详情，通常会列出发生冲突的行。

4.5 代码生成

工具生成的源代码可包含在应用中，以初始化引脚路由。源代码在更改时自动生成或通过主菜单中选择“引脚 > 现在生成”进行立即更新。



图 4-26. 生成代码

生成的代码显示在右边窗口的“源代码”选项卡中。该窗口显示了所有生成的文件且每个文件都有对应的选项卡。



图 4-27. 源视图

可以将生成的代码复制粘贴至源文件。该视图可为各函数生成代码。除函数注释外，还可将工具配置存储为 YAML 格式。注释并不用于直接编辑，可在之后用于重新存储引脚配置。

```

源代码 寄存器
pin_mux.c pin_mux.h
* BE CAREFUL MODIFYING THIS COMMENT - IT IS YAML SETTINGS FOR THE PINS TOOL ***
*/

#include "fsl_common.h"
#include "fsl_port.h"
#include "pin_mux.h"

#define PCR_ODE_ENABLED          0x01u  /*!< Open Drain Enable: Open drain output
#define PCR_SRE_FAST            0x00u  /*!< Slew Rate Enable: Fast slew rate is c
#define PIN0_IDX                 0u     /*!< Pin number for pin 0 in a port */
#define PIN1_IDX                 1u     /*!< Pin number for pin 1 in a port */
#define PIN2_IDX                 2u     /*!< Pin number for pin 2 in a port */
#define PIN3_IDX                 3u     /*!< Pin number for pin 3 in a port */

/*
 * TEXT BELOW IS USED AS SETTING FOR THE PINS TOOL *****
BOARD_InitPins:
- options: {coreID: single-core, enableClock: 'true'}
- pin_list:
  - {pin_num: '1', peripheral: GPIOE, signal: 'GPIO, 0', pin_signal: ADC1_SE4a/PTE0/SPI1
    slew_rate: fast, open_drain: enable}
  - {pin_num: '2', peripheral: GPIOE, signal: 'GPIO, 1', pin_signal: ADC1_SE5a/PTE1/LLWU
  - {pin_num: '3', peripheral: GPIOE, signal: 'GPIO, 2', pin_signal: ADC0_DP2/ADC1_SE6a/
  - {pin_num: '4', peripheral: GPIOE, signal: 'GPIO, 3', pin_signal: ADC0_DM2/ADC1_SE7a/
* BE CAREFUL MODIFYING THIS COMMENT - IT IS YAML SETTINGS FOR THE PINS TOOL ***
*/

/*FUNCTION*****
 *
 * Function Name : BOARD_InitPins
 * Description   : Configures pin routing and optionally pin electrical features.
 *
 *END*****/
void BOARD_InitPins(void) {
    CLOCK_EnableClock(kCLOCK_PortE);          /* Port E Clock Gate Contro
    PORT_SetPinMux(PORTE, PIN0_IDX, kPORT_MuxAsGpio);          /* PORTE0 (pin 1) is config
    PORTE->PCR[0] = ((PORTE->PCR[0] &
        (~(PORT_PCR_SRE_MASK | PORT_PCR_ODE_MASK | PORT_PCR_ISF_MASK))) /* Mask bits to zero
        | PORT_PCR_SRE(PCR_SRE_FAST)          /* Slew Rate Enable: Fast s
        | PORT_PCR_ODE(PCR_ODE_ENABLED)      /* Open Drain Enable: Open
    );
    PORT_SetPinMux(PORTE, PIN1_IDX, kPORT_MuxAsGpio);          /* PORTE1 (pin 2) is config
    PORT_SetPinMux(PORTE, PIN2_IDX, kPORT_MuxAsGpio);          /* PORTE2 (pin 3) is config
    PORT_SetPinMux(PORTE, PIN3_IDX, kPORT_MuxAsGpio);          /* PORTE3 (pin 4) is config

```

图 4-28. 生成代码

YAML 配置包含各引脚的配置。仅存储非默认值。



就多核处理器而言，各核都会生成源文件。

4.5.1 导出源代码

可使用导出向导来导出生成的源文件。

启动导出向导：

1. 在主菜单中选择“文件 > 导出”。
2. 选择“导出源文件”选项。

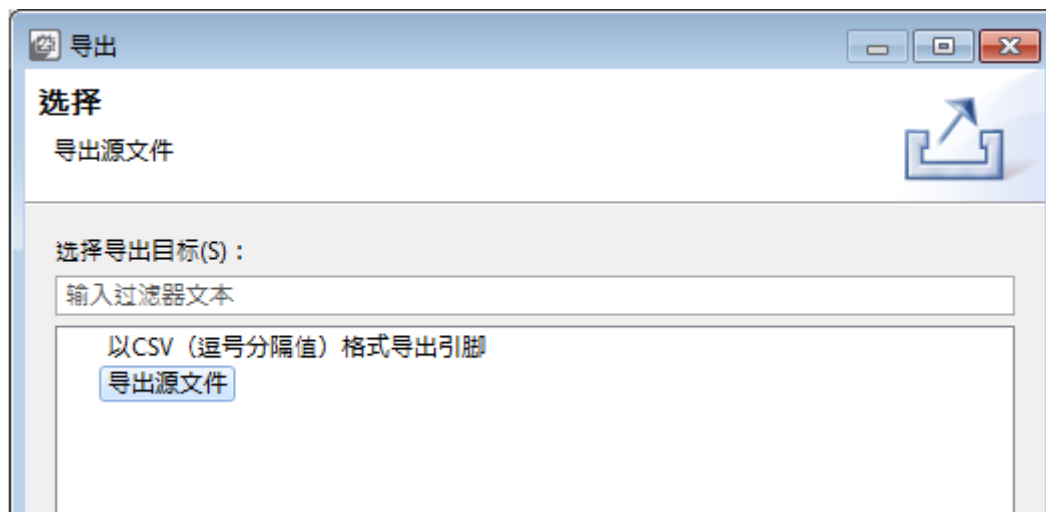


图 4-29. 导出向导

3. 单击“下一步”。
4. 选择需要存储生成文件的目标文件夹。

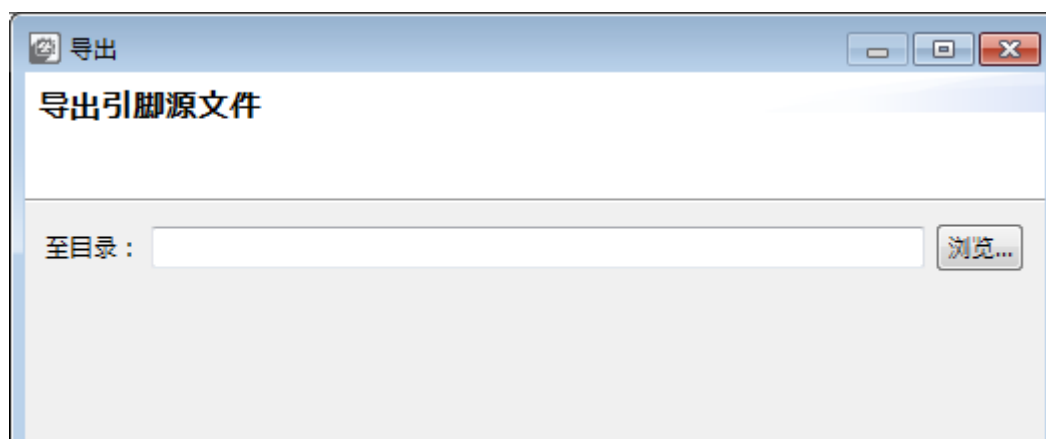


图 4-30. 选择目标文件夹

5. 单击“完成”。

* 网页版可以下载包含所有生成文件的 zip 文件。

4.5.2 导入源代码

导入源代码文件：

1. 在主菜单中选择”文件 > 导入“。

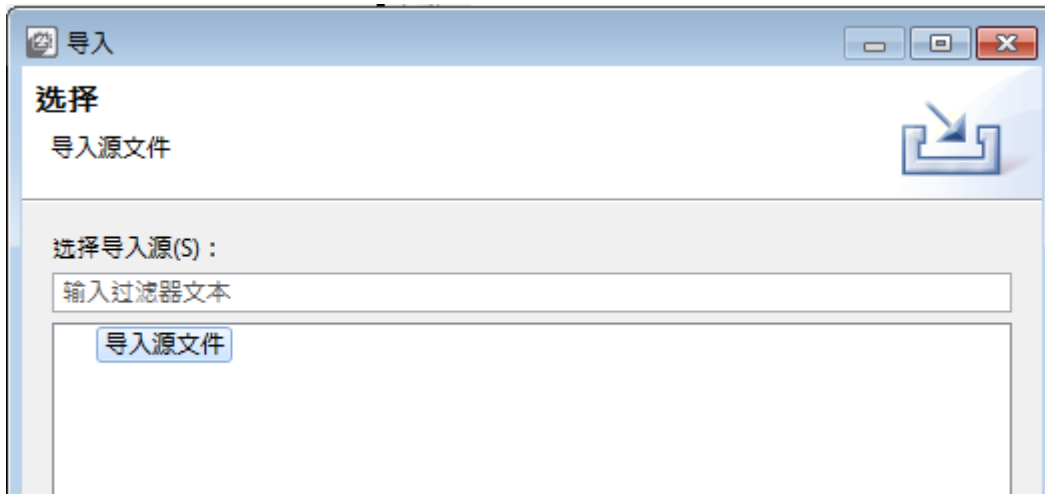


图 4-31. 导入向导

2. 选择”导入源文件“选项。
3. 单击”下一步“。
4. 可选择一个或多个 C 文件，使用”导入引脚源文件“对话框中的”浏览“按钮导入。

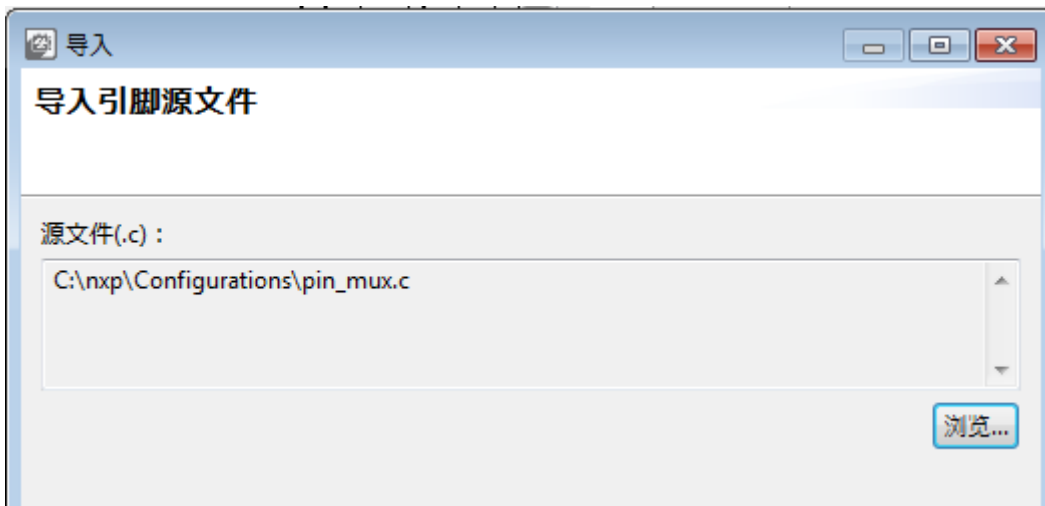


图 4-32. 导入引脚源文件

5. 单击”完成“。
- 所有文件都整合至当前配置。仅导入所有函数。若导入的函数与现有配置名称相同，则自动重命名为有索引的 1。例如，如果配置中已存在 BOARD_InitPins，则导入的函数重命名为 BOARD_InitPins1。

!	仅带有效 YAML 配置的 C 文件才能导入。仅导入配置，随后整个 C 文件将会根据设置重新创建（C 文件其他部分将被忽略）。
---	---

4.6 选项

4.6.1 配置首选项

为配置首选项，在主菜单中选择“编辑 > 首选项”。此时将出现配置首选项对话框。

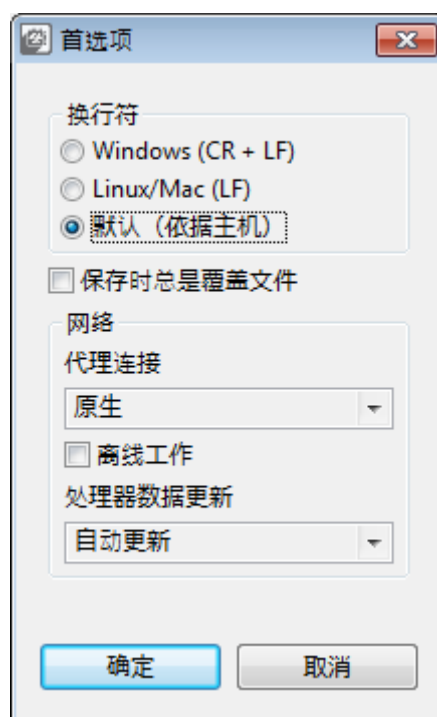


图 4-33. 配置首选项对话框

在此对话框中可以设置：

- 换行符（Windows、Linux 或主机默认系统）
- 覆盖保存
- 代理连接
 - 直接 – 不用代理直接连接网络
 - 原生 – 使用系统代理配置连接网络
- 离线模式

- 如果此功能已使能，则它不会从恩智浦云中下载/更新新数据，也不会显示所有可供下载的处理器。
- 处理器数据更新选项
 - 更新会以自动或手动（用户必须在请求时予以确认）的方式进行，或者是完全禁用。

4.6.2 引脚属性

如需设置引脚属性，在主菜单中选择“引脚 > 属性”。显示“引脚属性”对话框。



图 4-34. 引脚属性

通过该对话框，可以配置几个函数和代码生成选项。选定函数可使用各种设置，具体由选择的适当选项卡指定。可以在源代码中指定生成函数的名称、选定的内核*、使能时钟门控的选项*，或者写入函数说明（该说明将会生成在 C 文件中）。¹

4.6.3 软件更新

如需检查更新，选择“帮助 > 检查”获取更新菜单。该操作将连接服务器，检查是否有新版本可用。

1. *如果处理器支持的话

注

如需检查更新，则需要互联网连接。

第 5 章 高级功能

5.1 切换处理器

可切换当前配置中的处理器或封装。切换处理器或封装后，所有函数和已配置的引脚路由保持不变，但会应用到不同的处理器上。在更改到类似封装时，这个功能尤其有用。如果切换到完全不同的处理器，则可能产生很多冲突，或者引脚路由不可访问。在这种情况下，必须进入引脚路由表，并重新配置所有报错或报告冲突的引脚。

选择“文件 > 切换处理器”菜单，以在选定配置中更改处理器。

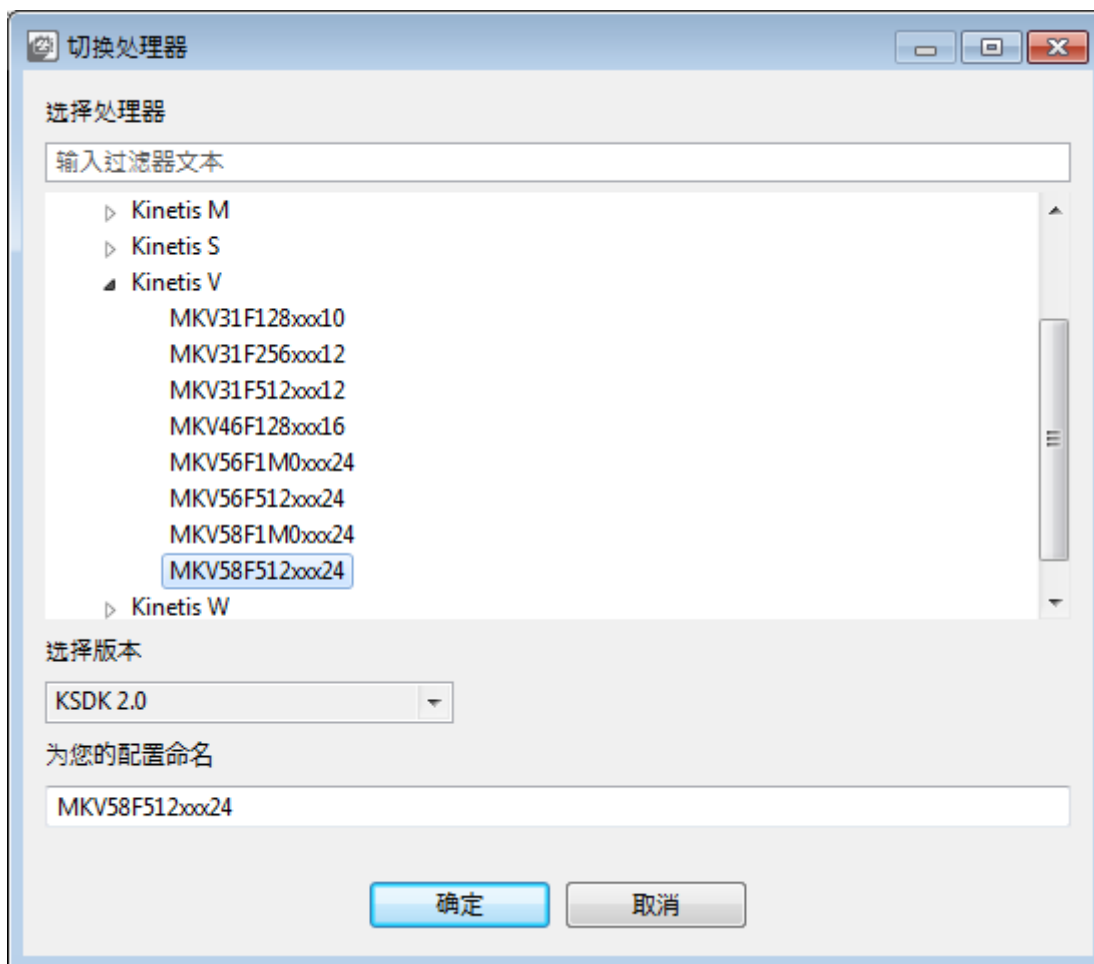


图 5-1. 切换配置

选择“文件 > 切换封装”菜单，以更改当前处理器的封装。



图 5-2. 切换封装

注

如需在网页版中切换处理器，则需要选择顶部的配置菜单，然后进行更改。

5.2 导出引脚表

要导出引脚表：

1. 在主菜单中选择“文件 > 导出”。
2. 在“导出”对话框中，选择“以 CSV（逗号分隔符）格式导出引脚”选项。



图 5-3. 导出对话框

3. 单击“下一步”。
4. 选择文件夹并指定希望导出的文件名。

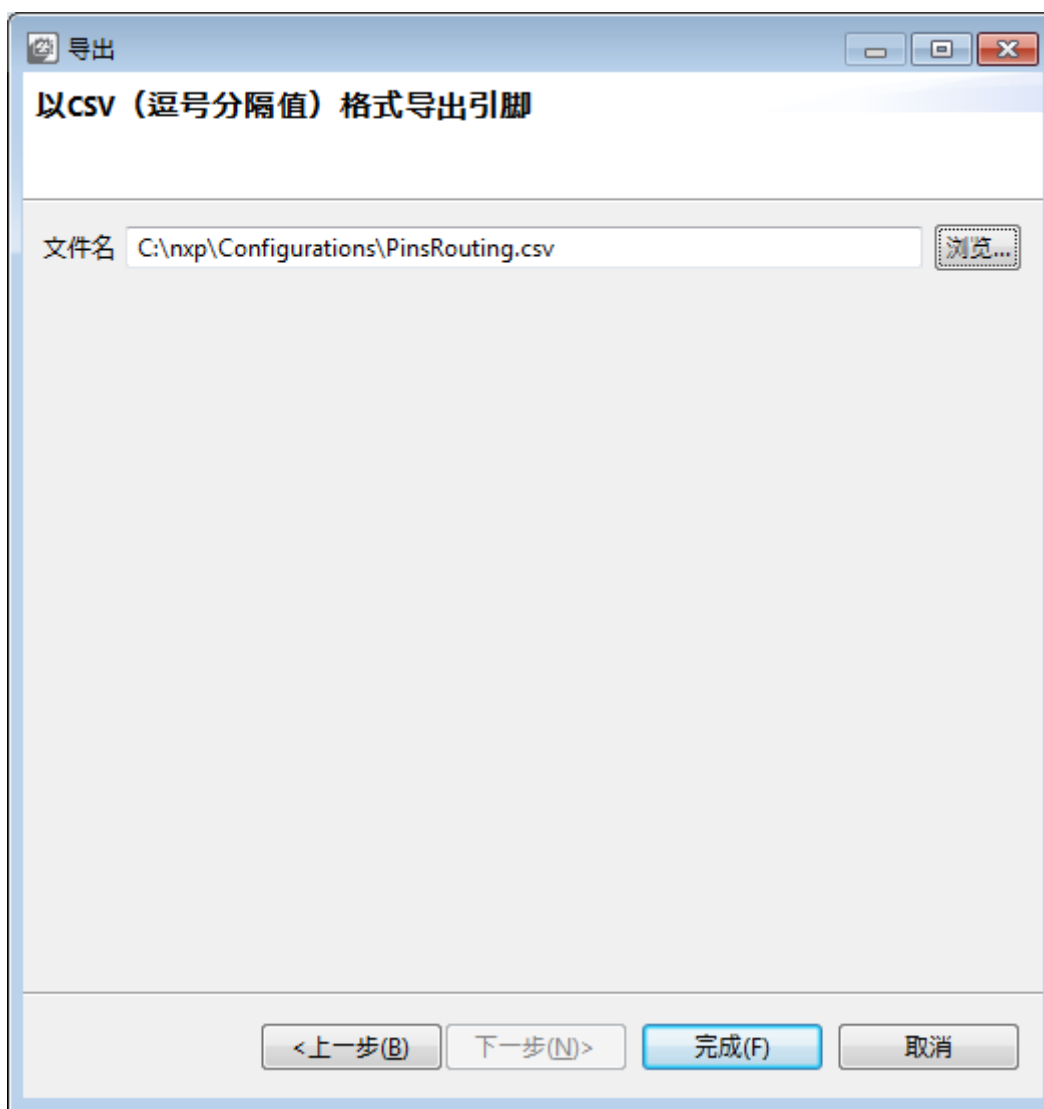


图 5-4. 以 CSV 格式导出引脚

5. 导出的文件中包含当前引脚视图的内容，以及功能和选定路由引脚的列表。

```

sep=;
Pin;Pin name;GPIO;FTM;ADC;UART;SPI;I2S;LLWU;I2C;CMP;SUPPLY;LPUART;USB;SIM;JTAG;RTC;EWM;Other;Routing for BOARD_InitPins
A1;PTE0/CLKOUT32K;PTE0/CLKOUT32K(GPIOE,GPIO,0);;ADC1_SE4a(ADC1,SEa,4);UART1_TX(UART1,TX);SPI1_PCS1(SPI1,PCS1);;I2C1_SDA(I2C1,SDA);;PTE0
B1;PTE1/LLWU_PO;PTE1/LLWU_PO(GPIOE,GPIO,1);;ADC1_SE5a(ADC1,SEa,5);UART1_RX(UART1,RX);SPI1_SOUT(SPI1,SOUT)/SPI1_SIN(SPI1,SIN);;PTE1/LLWU_PO(
C1;PTD5;PTD5(GPIOD,GPIO,5);FTM0_CH5(FTM0,CH,5);ADC0_SE6b(ADC0,SEb,6);UART0_CTS_b(UART0,CTS);SPI0_PCS2(SPI0,PCS2)/SPI1_SCK(SPI1,SCK);;
D1;USB0_DM;USB0_DM(USB0,DM);;
E1;USB0_DP;USB0_DP(USB0,DP);;
F1;ADC0_DM0/ADC1_DM3;ADC0_DM0/ADC1_DM3(ADC0,DM,0)/ADC0_DM0/ADC1_DM3(ADC0,SE,19)/ADC0_DM0/ADC1_DM3(ADC1,DM,3);;ADC0_DM0/ADC1_
G1;ADC0_DP0/ADC1_DP3;ADC0_DP0/ADC1_DP3(ADC0,DP,0)/ADC0_DP0/ADC1_DP3(ADC0,SE,0)/ADC0_DP0/ADC1_DP3(ADC1,DP,3)/ADC0_DP0/ADC1_DP3(ADC1,SE,3);
H1;VREF_OUT/CMP1_INS/CMP0_INS/ADC1_SE18;VREF_OUT/CMP1_INS/CMP0_INS/ADC1_SE18(ADC1,SE,18);;VREF_OUT/CMP1_INS/CMP0_INS/ADC1_SE18(CMP1,I
A2;PTD7/UART0_TX/FTM0_CH7/FTM0_FLT1/SPI1_SIN;PTD7(GPIOD,GPIO,7);FTM0_CH7(FTM0,CH,7)/FTM0_FLT1(FTM0,FLT,1);;UART0_TX(UART0,TX);SPI1_SIN(SPI1
B2;ADC0_SE7b/PTD6/LLWU_P15/SPI0_PCS3/UART0_RX/FTM0_CH6/FTM0_FLT0/SPI1_SOUT;PTD6/LLWU_P15(GPIOD,GPIO,6);FTM0_CH6(FTM0,CH,6)/FTM0_FLT0(FTM0,F
C2;PTD2/LLWU_P13/SPI0_SOUT/UART2_RX/LPUART0_RX/I2C0_SCL;PTD2/LLWU_P13(GPIOD,GPIO,2);;UART2_RX(UART2,RX);SPI0_SOUT(SPI0,SOUT);;PTD2/LLWU_P1
D2;VREGIN;VREGIN(USB0,VREGIN);;
E2;VOUT33;VOUT33(USB0,VOUT33);;
F2;ADC1_DM0/ADC0_DM3;ADC1_DM0/ADC0_DM3(ADC1,DM,0)/ADC1_DM0/ADC0_DM3(ADC1,SE,19)/ADC1_DM0/ADC0_DM3(ADC0,DM,3);;
G2;ADC1_DP0/ADC0_DP3;ADC1_DP0/ADC0_DP3(ADC1,DP,0)/ADC1_DP0/ADC0_DP3(ADC1,SE,0)/ADC1_DP0/ADC0_DP3(ADC0,DP,3)/ADC1_DP0/ADC0_DP3(ADC0,SE,3);
H2;DAC0_OUT/CMP1_IN3/ADC0_SE23;DAC0_OUT/CMP1_IN3/ADC0_SE23(ADC0,SE,23);;DAC0_OUT/CMP1_IN3/ADC0_SE23(CMP1,IN,3);;DAC0_OUT/CMP1_I
A3;PTD4/LLWU_P14/SPI0_PCS1/UART0_RTS_b/FTM0_CH4/EWM_IN/SPI1_PCS0;PTD4/LLWU_P14(GPIOD,GPIO,4);FTM0_CH4(FTM0,CH,4);;UART0_RTS_b(UART0,RTS);SP
B3;PTD3/SPI0_SIN/UART2_TX/LPUART0_TX/I2C0_SDA;PTD3(GPIOD,GPIO,3);;UART2_TX(UART2,TX);SPI0_SIN(SPI0,SIN);;I2C0_SDA(I2C0,SDA);;LPUART0_TX(
C3;PTD0/LLWU_P12;PTD0/LLWU_P12(GPIOD,GPIO,0);;UART2_RTS_b(UART2,RTS);SPI0_PCS0(SPI0,PCS0/SS);;PTD0/LLWU_P12(LLWU,WAKEUP,P12);;LPUART0_RT
D3;PTA0/UART0_CTS_b/FTM0_CH5/JTAG_TCLK/SWD_CLK/EZP_CLK;PTA0(GPIOA,GPIO,0);FTM0_CH5(FTM0,CH,5);;UART0_CTS_b(UART0,CTS);;JTAG_TCLK(JT

```

图 5-5. 导出的文件内容

导出的内容可在其他工具中使用，以做进一步处理。例如，在与下图中的数据块对齐之后进行查看。

```

sep=;
Pin :Pin name                                ;GPIO                                ;FTM                                ;ADC
A1 :PTF0/CLKOUT32K                          ;PTF0/CLKOUT32K (GPIOE, GPIO, 0) ;                                ;ADC1_SE4a (ADC1, SEa, 4)
B1 :PTE1/LLWU_P0                             ;PTE1/LLWU_P0 (GPIOE, GPIO, 1)  ;                                ;ADC1_SE5a (ADC1, SEa, 5)
C1 :PTD5                                      ;PTD5 (GPIOD, GPIO, 5)         ;FTM0_CH5 (FTM0, CH, 5)         ;ADC0_SE6b (ADC0, SEb, 6)
D1 :USBD0_DM                                 ;                                ;                                ;                                ;
E1 :USBD0_DP                                 ;                                ;                                ;                                ;
F1 :ADCO_DMO/ADC1_DMS                       ;                                ;                                ;ADCO_DMO/ADC1_DMS (ADC0, DM, 0)/ADCO_DMO/ADC
G1 :ADCO_DFO/ADC1_DFP3                     ;                                ;                                ;ADCO_DFO/ADC1_DFP3 (ADC0, DP, 0)/ADCO_DFO/ADC
H1 :VREF_OUT/CMF1_INS/CMF0_INS/ADC1_SE18    ;                                ;                                ;VREF_OUT/CMF1_INS/CMF0_INS/ADC1_SE18 (ADC1
A2 :PTD7/UART0_TX/FTM0_CH7/FTM0_FLT1/SP11_SIN ;PTD7 (GPIOD, GPIO, 7)         ;FTM0_CH7 (FTM0, CH, 7)/FTM0_FLT1 (FTM0, FLT, 1) ;                                ;
B2 :ADCO_SE7b/PTD6/LLWU_P15/SPI0_PCS3/UART0_RX/FTM0_CH6/FTM0_FLT0/SP11_SOUT;PTD6/LLWU_P15 (GPIOD, GPIO, 6) ;FTM0_CH6 (FTM0, CH, 6)/FTM0_FLT0 (FTM0, FLT, 0)/FTM0_FLT0 (FTM0, TRG, 2);ADCO_SE7b (ADC0, SEb, 7)
C2 :PTD2/LLWU_P13/SPI0_SOUT/UART2_RX/LPUART0_RX/I2C0_SCL ;PTD2/LLWU_P13 (GPIOD, GPIO, 2) ;                                ;                                ;
D2 :VREG3M3                                 ;                                ;                                ;                                ;
E2 :VOUT33                                  ;                                ;                                ;                                ;
F2 :ADC1_DMO/ADCO_DMS                      ;                                ;                                ;ADCO_DMO/ADCO_DMS (ADC1, DM, 0)/ADC1_DMO/ADC
G2 :ADC1_DFO/ADCO_DFP3                    ;                                ;                                ;ADCO_DFO/ADCO_DFP3 (ADC1, DP, 0)/ADC1_DFO/ADC
H2 :DAC0_OUT/CMF1_INS/ADCO_SE23            ;                                ;                                ;DAC0_OUT/CMF1_INS/ADCO_SE23 (ADC0, SE, 23)
A3 :PTD4/LLWU_P14/SPI0_PCS1/UART0_RTS_b/FTM0_CH4/ENM_IN/SP11_PCS0 ;PTD4/LLWU_P14 (GPIOD, GPIO, 4) ;FTM0_CH4 (FTM0, CH, 4)         ;                                ;
E3 :PTD3/SPI0_SIN/UART2_TX/LPUART0_TX/I2C0_SDA ;PTD3 (GPIOD, GPIO, 3)         ;                                ;                                ;
C3 :PTD0/LLWU_P12                          ;PTD0/LLWU_P12 (GPIOD, GPIO, 0) ;                                ;                                ;
D3 :PTA0/UART0_CTS_b/FTM0_CH5/JTAG_TCLKR/SWD_CLK/E2P_CLK ;PTA0 (GPIOA, GPIO, 0)         ;FTM0_CH5 (FTM0, CH, 5)         ;                                ;
E3 :VSS80                                  ;                                ;                                ;                                ;
F3 :VSSA                                  ;                                ;                                ;VSSA (ADC0, SE, 30)/VSSA (ADC1, SE, 30)/VSSA (AI
H3 :VREFL                                  ;                                ;                                ;VREFL (ADC0, SE, 30)/VREFL (ADC1, SE, 30)/VREFL
H3 :XNAL32                                  ;                                ;                                ;                                ;
A4 :ADCO_SE5b/PTD1/SPI0_SCK/UART2_CTS_b/LPUART0_CTS_b ;PTD1 (GPIOD, GPIO, 1)         ;                                ;ADCO_SE5b (ADC0, SEb, 5)
B4 :ADC1_SE6b/FTC10/I2C1_SCL/I2S0_RX_FS    ;FTC10 (GPIOC, GPIO, 10)       ;                                ;ADC1_SE6b (ADC1, SEb, 6)
C4 :VSS9                                  ;                                ;                                ;                                ;
D4 :PTA1/UART0_RX/FTM0_CH6/JTAG_TDI/E2P_DI ;PTA1 (GPIOA, GPIO, 1)         ;FTM0_CH6 (FTM0, CH, 6)         ;                                ;
E4 :VDD81                                  ;                                ;                                ;                                ;
F4 :VDDA                                  ;                                ;                                ;VDDA (ADC0, SE, 29)/VDDA (ADC1, SE, 29)/VDDA (AI
G4 :VREFH                                  ;                                ;                                ;VREFH (ADC0, SE, 29)/VREFH (ADC1, SE, 29)/VREFH
    
```

图 5-6. 与数据块对齐

第 6 章 技术支持

如有任何疑问或需要帮助，请在 Kinetis 或 i.MX 社区进行搜索或发布新问题。访问 <https://community.freescale.com/community/kinetis> 或 <https://community.freescale.com/community/imx>。



How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

本文档中的信息仅供系统和软件实施方使用 Freescale 产品。本文并未明示或者暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。Freescale 保留对此处任何产品进行更改的权利，恕不另行通知。

Freescale 对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用程序或者使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于后果性的或附带性的损害在内的所有责任。Freescale 的数据表和/或规格中所提供的“典型”参数在不同应用中可能并且确实不同，实际性能会随时间而有所变化。所有运行参数，包括“经典值”在内，必须经由客户的技术专家对每个客户的应用程序进行验证。Freescale 未转让与其专利权及其他权利相关的许可。

Freescale 销售产品时遵循以下网址中包含的标准销售条款和条件：freescale.com/SalesTermsandConditions。

Freescale, the Freescale logo, and Kinetis, are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners.

© 2016 Freescale Semiconductor, Inc.

© 2016 飞思卡尔半导体有限公司

